

Міністерство освіти і науки України

Хмельницький обласний інститут післядипломної педагогічної освіти

Збірник олімпіадних задач з інформатики у Хмельницькій області

Упорядники:

Старший викладач

кафедри менеджменту та освітніх
технологій

Ребрина В.А.

Вчитель інформатики Кам'янець-
Подільського ліцею

Аносов Д.О.

Хмельницький, 2011

Зміст

Вступ.....	3
Геометрія.....	5
Графічні задачі.....	20
Динамічне програмування.....	21
Довга арифметика.....	56
Жадібний алгоритм.....	61
Задачі для початківців.....	62
Комбінаторика.....	72
Масиви.....	81
Математика.....	107
Математичне моделювання.....	123
Обробка рядків.....	132
Послідовності.....	155
Рекурсія, перебір.....	165
Сортування.....	188
Структури даних.....	192
Теорія графів.....	193
Теорія ігор.....	217
Теорія чисел.....	238
Фізика.....	261
Різне.....	262

Вступ

На відміну від звичайних програм, що створюються програмістами повсякденно, клас олімпіадних задач досить вузький, але практичний з точки зору критеріїв виявлення здібностей учнів програмувати. Як правило, олімпіадна задача представляє собою деяку проблему, для вирішення якої потрібно використовувати свій IQ майже на межі, однак, сам текст програми може бути зовсім незначним і розміщуватися на одній сторінці.

Не існує єдиного методу розв'язку олімпіадних задач. Навпаки, кількість методів постійно поповнюється. Деякі задачі можна розв'язати кількома різними методами або комбінацією методів. Характерна особливість олімпіадних задач у тому, що розв'язок на перший погляд нескладної проблеми може вимагати застосування методів, які використовуються в серйозних математичних дослідженнях.

Учасниками олімпіад та членами журі часто вводиться неофіційна класифікація завдань олімпіадного програмування. Це пов'язано з тим, що, незважаючи на відмінність сюжетів, алгоритми та структури даних, що використовуються для вирішення завдань часто схожі один на одного і їх можна розділити на кілька категорій. Категорії зазвичай носять назви класів алгоритмів або відповідних розділів математики. Необхідно відзначити, що завжди можливо придумати завдання, що не відноситься ні до однієї з категорій.

Необхідно відзначити, що завжди можливо придумати завдання, що не відноситься ні до однієї з категорій. Для цього вводять категорію «Різне»

На кожен категорію можна придумати завдання самого різного рівня складності. Тому факт віднесення завдання до однієї з категорій не може свідчити про її складність.

Основними категоріями олімпіадних задач є:

- Геометрія
- Графічні задачі
- Динамічне програмування
- Довга арифметика
- Жадібний алгоритм
- Задачі для початківців
- Комбінаторика
- Масиви
- Математика
- Математичне моделювання
- Обробка рядків
- Послідовності
- Рекурсія, перебір
- Різне
- Сортування
- Структури даних
- Теорія графів
- Теорія ігор
- Теорія чисел
- Фізика

Геометрія

Задача 1.

(II обласна олімпіада, 1988 рік)

Вершини ламаної мають координати (X, Y) , де $Y=1+X/2$, $X=0,1,2,3,4,5$. Скласти програму обчислення площі фігури, обмеженої ламаною лінією, віссю абсцис і прямими $X=0$, $X=5$.

Задача 2.

(III обласна олімпіада, 1989 рік)

На площині дано координати чотирьох точок. Три з них утворюють трикутник. Перевірити, чи є внутрішньою для трикутника четверта крапка.

Задача 3.

(IV обласна олімпіада, 1990 рік)

На прямій пофарбували n відрізків. Відомі координати $L [i]$ лівого кінця і координати $R [i]$ правого кінця i -го відрізка ($i = 1, 2, \dots, n$). Знайти суму довжин пофарбованої частини прямої.

Задача 4. «Одна чверть»

(VI обласна олімпіада, 1992 рік)

Дві точки на площині, що не лежать на координатних осях, задані своїми координатами: $A(X_1, Y_1)$ і $B(X_2, Y_2)$. Перевірити, чи лежать ці точки в одній координатній чверті.

Задача 5. «Опуклість многокутника»

(VI обласна олімпіада, 1992 рік)

Відомі координати вершин многокутника (цілі числа). Чи є цей многокутник опуклим?

Задача 6. «Трикутна піраміда»

(VII обласна олімпіада, 1993 рік)

Задано координати вершин S , A , B і C трикутної піраміди $SABC$.
Визначити, які точки потрібно з'єднувати штриховою лінією при побудові її графічного зображення.

Задача 7. «Відрізок»

(XII обласна олімпіада, 1998 рік)

Задане число N - кількість точок на площині, та числа $X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N$ – координати точок. Потрібно скласти програму `LINE.*`, яка визначає координати кінців відрізка найменшої довжини та довжину відрізка. На екрані зобразити точки і шуканий відрізок.

Задача 8. «Куски»

(XIII обласна олімпіада, 1999 рік)

На площині задано N прямих. Напишіть програму, яка визначає, на скільки кусків розбивають площину ці прямі.

Задача 9. «Прямокутники»

(XIV обласна олімпіада, 2000 рік)

На екрані EOM по черзі зображають N ($N < 10000$) прямокутників, причому колір кожного наступного прямокутника відрізняється від кольору фону і кольорів вже побудованих прямокутників. Складіть програму `RECTANGL*`, яка визначає скільки пікселів (точок екрану) матимуть колір першого прямокутника по закінченні побудови.

Вхідні дані

У першому рядку текстового ASCII файлу `RECTANGL.DAT` записане ціле число N ; у наступних N рядках записані через пропуск по чотири цілих числа (X_1, Y_1, X_2, Y_2) - координати двох протилежних вершин кожного із прямокутників.

Вихідні дані

У текстовий ASCII-файл RECTANGL.SOL потрібно записати єдине ціле число - кількість пікселів.

Приклад вхідних і вихідних даних:

```
RECTANGL.DAT          RECTANGL.SOL
4                      18
1 1 5 6
5 3 7 4
2 4 3 5
3 2 5 4
```

Задача 10. «Прямокутники»

(XVI обласна олімпіада, 2002 рік)

На площині задано N прямокутників. Сторони кожного прямокутника паралельні осям координат. Визначити площу тієї частини площини, яка належить усім прямокутникам.

Вхідні дані:

У першому рядку число N, у наступних N рядках містяться по 4 числа X1 Y1 X2 Y2 - координати лівого верхнього та правого нижнього кутів прямокутника відповідно. Координати - цілі числа, що лежать у межах від -10000 до 10000 ($1 \leq N \leq 1000$).

Вихідні дані:

єдине число - значення площі.

Приклад:

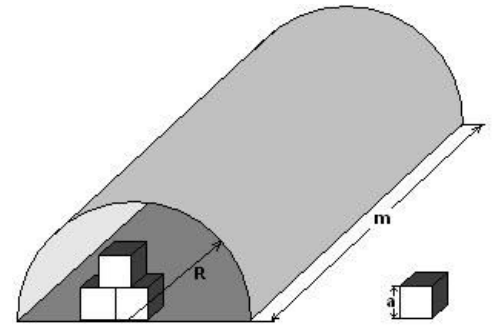
```
2
10 40 30 20
20 30 40 10
Відповідь:
100
```

Задача 11. «Заощадливий Дід Мороз»

(XX обласна олімпіада, 2006 рік)

Дід Мороз, готуючись до Нового року, вирішив наперед застатись подарунками для дітей і дорослих. Збирав він їх протягом цілого року і тому їх кількість може бути дуже великою.

Є лише проблема їх зберігання. Для цього Дід Мороз має в своєму розпорядженні сховище напівциліндричної форми (див. мал.) довжиною m і радіусом R одиниць.



Самі ж подарунки являють собою коробки у вигляді куба з ребром a одиниць.

Зрозуміло, що Дід Мороз має бажання заощадити якомога більше подарунків. Конструкцією сховища передбачено, що складати їх можна лише так, щоб одне ребро коробки було паралельним осі сховища.

Знаючи розміри сховища m і R , допоможіть йому підрахувати кількість подарунків, які вміщуються у сховищі.

Вхідні дані: із стандартного вхідного потоку вводиться три числа m , R , a , розділені одним пропуском ($0 < m \leq 1000000$, $0 < R \leq 1000000$, $0 < a < 1000000$).

Вихідні дані: у стандартний вихідний потік виведіть єдине ціле число – максимальну кількість подарунків.

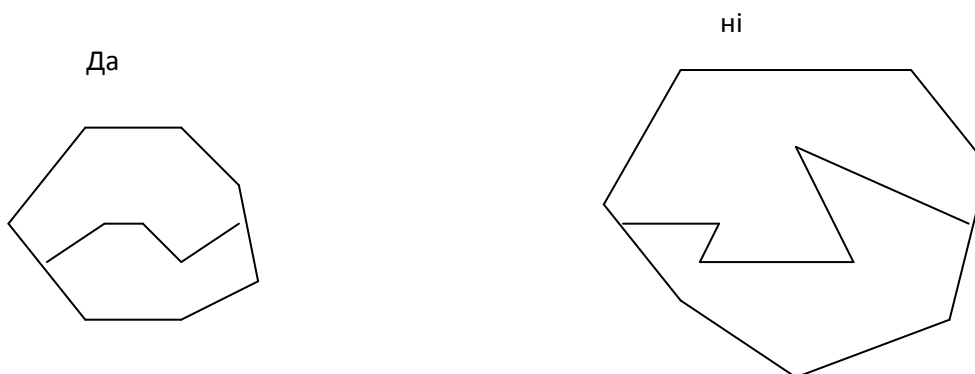
Вхідні дані	Вихідні дані
4 3 2	4
1000 1000 1	1569277000

Задача 12. «Розріз»

(Заочна олімпіада, 2000-2001)

Задано розріз плоскої пластини у вигляді несамопересічної ламаної. Вважаючи, що кінці ламаної лежать на краю пластини і ламана знаходиться

цілком усередині неї, дати відповідь на питання: "Чи можна роз'єднати отримані дві частини пластини, рухаючи їх у горизонтальній площині?"



Вхідні дані: у першому рядку міститься ціле число N -кількість точок ламаної. У другому рядку міститься $2N$ дробових чисел, розділених пробілами: $x_1 y_1 x_2 y_2 \dots$ - координати точок ламаної. $2 \leq N \leq 1000$, $-1000.0 \leq x_i, y_i \leq +1000.0$. Вихідні дані: у вихідний потік вивести символ "+", якщо роз'єм пластини можливий, інакше вивести "-".

Задача 13.

(Заочна олімпіада, 2000-2001)

На площині накреслили $1 \leq N \leq 10000$ відрізків. Необхідно знайти точку, яка належить найбільшому числу відрізків. Якщо така точка не одна, то видати будь-яке правильне рішення.

Вихідні дані: у першому рядку вхідного потоку число N , а в наступних N рядках по два дійсних числа (координати початку і кінця чергового відрізка). У перший рядок вихідного потоку вивести координату знайденої точки. Примітка: координати відрізків лежать в межах від -10000.00 до 10000.00

Задача 14. n -кутник

(Заочна олімпіада, 2001-2002)

n -кутник задано координатами (x, y) своїх вершин в порядку їх обходу ($-1000.0 \leq x, y \leq 1000.0$, $3 \leq n \leq 1000$). n -кутник не має самоперетинів.

Визначить за годинниковою стрілкою чи проти перераховано вершини багатокутника.

В першій стрічці вхідного потоку міститься число n . В n наступних стрічках містяться пари чисел x, y розділені пропусками.

У вихідний потік вивести один символ: '>', якщо багатокутник задано за годинниковою стрілкою, '<', якщо проти.

Приклад.

Вхід:

5

-3.7 2.2

1 1.4

3.3 3.0

2.5 -4

-3.7 -4

Вихід: >

Задача 15. «Трикутний більярд»

(Заочна олімпіада, 2002-2003)

В більярдному клубі "Бермудський трикутник" більярдні столи трикутної форми. Ціль гравця в такому більярді відбити кульку від всіх трьох сторін так, щоб вона повернулась в початкове положення (пройшла через точку старту). Нехай дано стіл з вершинами в точках A, B, C і точку O - точку старту. Вкажіть на стороні AB точку, в яку необхідно виконати удар, щоб відбившись послідовно від сторін AB, BC, AC кулька повернулась в точку O .

Технічні умови.

Вхідний файл BILLIARD.DAT містить чотири рядки:

- в першому через пропуск координати (x, y) точки A ,
- в другому через пропуск координати (x, y) точки B ,
- в третьому через пропуск координати (x, y) точки C ,

- в четвертому через пропуск координати (x;y) точки O.

Вихідний файл BILLIARD.SOL містить єдиний рядок, в якому через пропуск записані два числа: координати точки з точністю до двох знаків після коми.

Задача 16. «НМС-покриття»

(Заочна олімпіада, 2003-2004)

Деяка гіпотетична фірма НМС - оператор мобільного зв'язку, вирішила "покрити" зв'язком N населених пунктів, розташованих вздовж магістралі. З метою економії коштів ставиться один передавач, обов'язково на магістралі. Допоможіть інженерам знайти на магістралі місце для розташування передавача, щоб покрити всі населені пункти і при цьому мати найменший радіус дії.

Технічні умови:

кількість населених пунктів N;

розташування населених пунктів задається парами дійсних чисел (x,y) в прямокутній декартовій системі координат;

магістраль співпадає з віссю Ox.

Вхідні дані знаходяться в файлі "COVERING.DAT", який має структуру:

в першому рядку єдине число N ($0 < N \leq 10000$);

в наступних N рядках по два дійсних числа розділених пропуском - $10000 \leq x \leq 10000$, $-10000 \leq y \leq 10000$.

Результат вивести у файл "COVERING.SOL", який має структуру:

в одному рядку через пропуск записано два дійсних числа: абсциса точки розташування передавача, та радіус дії. Результати вивести заокругливши до тисячних.

Приклад 1:

```
COVERING.DAT      COVERING.SOL
2                0.000 5.000
0 0
0 5
```

Приклад 2:

```
COVERING.DAT      COVERING.SOL
2                2.500 2.500
0 0
5 0
```

Приклад 3:

```
COVERING.DAT      COVERING.SOL
4                1.298 11.103
1.23 4.56
7.8 9
-1.306 -4.05
-7.4 6.9
```

Задача 17. «Перетин трикутників»

(Заочна олімпіада, 2003-2004)

Визначіть, чи перетинаються два трикутники A і B , задані координатами своїх вершин $(A_x1;A_y1)$, $(A_x2;A_y2)$, $(A_x3;A_y3)$ та $(B_x1;B_y1)$, $(B_x2;B_y2)$, $(B_x3;B_y3)$ відповідно. Всі координати - цілі числа в діапазоні від -10000 до +10000.

Вершини трикутників та їх сторони гарантовано не співпадають.

Примітка: рахується, що трикутники перетинаються, якщо на площині існує така точка, що знаходиться всередині обох трикутників.

Вхідні дані:

В першій стрічці вхідного потоку міститься 6 цілих чисел розділених пропусками - координати першого трикутника:

$Ax_1 Ay_1 Ax_2 Ay_2 Ax_3 Ay_3$.

В другій стрічці вхідного потоку міститься ще 6 чисел - координати другого трикутника:

$Vx_1 Vy_1 Vx_2 Vy_2 Vx_3 Vy_3$.

Вихідні дані:

У вихідний потік слід вивести символ "+", якщо трикутники перетинаються, або символ "-", якщо ні.

Приклад:

Вхідний потік:

0 2 0 6 4 2

1 0 5 0 1 4

Вихідний потік:

+

Задача 18. «Прямокутник»

(Заочна олімпіада, перша ліга, 2005-2006)

Сторони прямокутника паралельні координатним осям (вісь X спрямована вправо, вісь Y - вгору). Відомі координати кінців двох сторін. Всі координати знаходяться в межах від -100000 до 100000. Скласти програму, для визначення того, як розміщена кожна з цих сторін у прямокутнику (верхня, нижня, ліва чи права).

Вхідні дані: у текстовому файлі figure.dat - два рядки, в кожному з яких - по 4 цілі числа X_1, Y_1, X_2, Y_2 , відокремлені пропусками - координати кінців двох сторін прямокутника.

Вихідні дані: у одному рядку текстового файлу figure.sol - два символи, що позначають відповідні сторони прямокутника: W - верхня, N - нижня, L - ліва, P - права.

Приклад вхідних даних - текстовий файл figure.dat:

10 10 20 10

20 10 20 40

Приклад вихідних даних - текстовий файл - figure.sol:

NP

Задача 19. «Площа»

(Заочна олімпіада, перша ліга, 2005-2006)

Трикутник на площині задається цілочисельними координатами вершин. Для заданої точки $Z(X;Y)$ визначити, чи належить вона трикутнику.

Вхідні дані: В першому рядку текстового файлу **trukut.dat** записані через пропуск координати вершин: **X1 Y1 X2 Y2 X3 Y3**. У другому через пропуск координати т. **Z : X Y**.

Вихідні дані: В текстовий файл **trukut.sol** записати «yes» у разі коли т. $Z(X;Y)$ належить трикутнику, або «no» в іншому випадку.

Приклад вхідних і вихідних файлів:

trukut.dat	trukut.sol
0 0 0 3 4 0	yes
1 1	

Задача 20. «Вертоліт»

(Заочна олімпіада, перша ліга, 2006-2007)

З вертольота потрібно скинути вантаж на прямокутну площадку, координати вершин якої відомі (на схемі позначена цифрою 0). В разі промаху вантаж може потрапити в одну з восьми областей поблизу прямокутника (на схемі - цифри від 1 до 8):

1	2	3
8	0	4
7	6	5

Пілот спостерігає за падінням і передає на наземну станцію номер області. При цьому границі прямокутника належать йому, а інші граничні лінії належать відповідним областям з парними номерами. Допоможіть

пілоту за відомими координатами вершин прямокутника та точки падіння вантажу визначити, яке число слід передати на станцію.

Вхідні дані: зі стандартного вхідного потоку програма зчитує координати лівого верхнього, правого верхнього, правого нижнього, лівого нижнього кутів прямокутника та координати точки падіння. Всі координати - цілі числа в межах від -100000 до 100000. Координати в кожній парі відокремлені пропуском і вводяться в окремому рядку.

Вихідні дані: у стандартний вихідний потік вивести одне ціле число - номер області, в яку потрапив вантаж.

Приклад 1	
вхідні дані	вихідні дані
1 1	3
2 1	
2 0	
1 0	
3 3	
Приклад 2	
вхідні дані	вихідні дані
2000 20000	0
4000 20000	
4000 0	
2000 0	
3000 1	

Задача 21. «Листівка та конверт»

(Заочна олімпіада, перша ліга, 2006-2007)

Задані розміри прямокутних листівки та конверта. Потрібно визначити чи поміститься листівка у конверті.

Обмеження: розміри листівки та конверта – цілі додатні числа, які не більші за 100.

Вхідний файл: *postcard.in*

У першому рядку записані розміри листівки, у другому розміри конверта (в обох випадках відокремлені пропусками).

Вихідний файл: *postcard.out*

Якщо листівку можна помістити у конверті, то вивести “Possible” , якщо ні – то “Impossible”

Приклад 1.

postcard.in	postcard.out
1 10 9 9	Possible

Приклад 2.

postcard.in	postcard.out
9 9 9 9	Impossible

Задача 22. «День студента»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Ім'я вхідного файлу: *s.in*

Ім'я вихідного файлу: *s.out*

17 листопада студенти святкували «День студента». Мешканці одного міста, які проживають біля студентського гуртожитку факультету кібернетики з жахом споминають той день і досі. Студент Степан напився малинового соку, включив на повну гучність музику авторства Верки

Сердючки і пус-тився в танок. Танок Степана приніс багато збитків присадибним клумбам і ділянкам...

Цього року комендант гуртожитку вирішили отримати користь із відпочинку студентів, на пустирі біля гуртожитку вирішено було організувати город, на якому наступним літом будуть рости помідори, огірки і капуста. Коменданту так сподобався танок Степана, що вона вирішила використати його для протоптування доріжок між грядками.

Пустир представляє собою площину, на якій введена прямокутна декартова система координат. Доріжка представляє собою відрізок на цій площині. Доріжки розбивають площину на деяку кількість частин. Будемо називати грядками ті із цих частин, які мають кінцеву площу.

Танок Степана описується набором із N точок $P(0)$, $P(1)$, ..., $P(N-1)$. Степан починає свій танок у точці $P(0)$ і далі буде послідовно переміщуватись у точки $P(1)$, $P(2)$, ..., $P(N-1)$. Між кожною парою сусідніх точок (тобто між точками $P(i)$ і $P(i+1)$ для $0 \leq i \leq N-2$) Степан протопче доріжку, що представляє собою відрізок, який з'єднує ці дві точки. Комендант гуртожитку хоче спрогнозувати, скільки грядок утвориться після виконання танцю. Допоможіть їй у цій нелегкій справі.

Вхідні дані:

У першому рядку число N , у першому з наступних двох рядків елементи масиву x , а у наступному рядку елементи масиву y . Масиви цілих чисел x і y містять рівно N елементів і описують координати точок $P(0)$, $P(1)$, ..., $P(N-1)$. Координати точки P_i рівні $(x[i], y[i])$ для $0 \leq i \leq N-1$.

Вихідні дані:

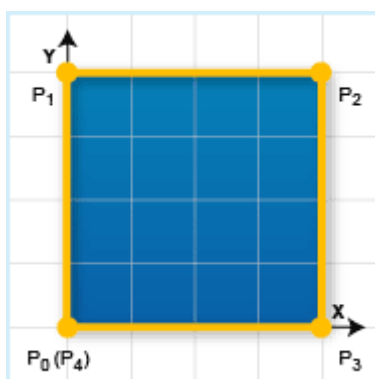
Ціле число, рівне кількості грядок на пустирі після виконання Степаном танцю, описаного у вхідних даних.

Обмеження:

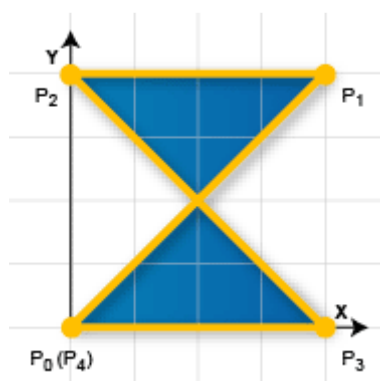
1. Число N від 5 до 100 включно.
2. Кожен елемент масиву x від -100 до 100 включно.
3. Кожен елемент масиву y від -100 до 100 включно.

Пояснення до прикладів:

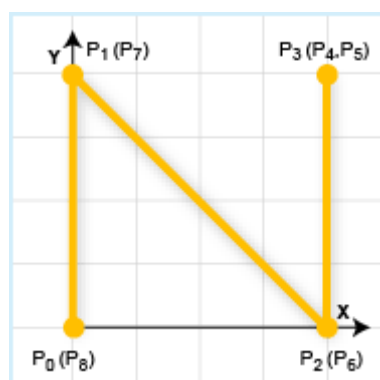
1. У результаті танцю Степана утворилась одна грядка квадратної форми



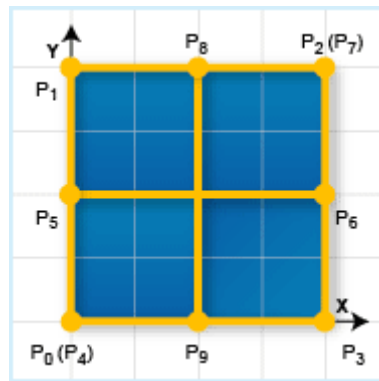
2. У результаті танцю Степана утворились дві грядки трикутної форми



3. Степан прогулявся зигзагом від точки $(0, 0)$ до точки $(4, 4)$, потоптався на місці, а потім точно так же вернувся назад у точку $(0, 0)$. У результаті не утворилось ні одної грядки.



4. У результаті танцю утворилось 4 квадратних грядки. Зверніть увагу, що початкова і кінцева точки танцю можуть не співпадати.



Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
5 0 0 4 4 0 0 4 4 0 0	1
5 0 4 0 4 0 0 4 4 0 0	2
9 0 0 4 4 4 4 4 0 0 0 4 0 4 4 4 0 4 0	0
10 0 0 4 4 0 0 4 4 2 2 0 4 4 0 0 2 2 4 4 0	4
10 3 -2 2 5 3 4 1 -2 3 4 2 -1 -3 4 0 -2 3 1 -2 5	9

Графічні задачі

Задача 1.

(II обласна олімпіада, 1988 рік)

Припустимо, що фігура "сніжинка" утворюється так: з центру виростає 6 кристаликів-відрізків довжини L під кутом 60 градусів і на кожному вільному кінці відрізка виростає по 5 нових відрізків, довжиною в k разів менше L , утворюють кути в 60 градусів між вільними від різкими. Аналогічно, з вільних кінців отриманих відрізків виростає по 5 нових відрізків, довжина яких в k разів менше довжини кристалика на попередньому рівні. Написати алгоритм, який малює "сніжинку" для будь-яких L, k, N .

Задача 2. «Вкладені квадрати»

(V обласна олімпіада, 1991 рік)

На площині задано систему вкладених квадратів з спільною вершиною в початку декартової системи координат таких, що одна з діагоналей кожного квадрата належить додатній півосі по Ox і площа кожного наступного квадрата в два рази менша площі попереднього. Складіть алгоритм малювання всіх квадратів цієї системи, площа яких не менша заданого числа.

Динамічне програмування

Задача 1.

(II обласна олімпіада, 1988 рік)

Стіни деякого спрощеного лабіринту можуть бути або паралельними, або перпендикулярними, а довжини цілочисельними (наприклад, лабіринт намальований на аркуші в клітину). Кільцеві коридори в лабіринті відсутні і з будь-якої його точки існує вихід. Автомат, що знаходиться в лабіринті, може виконувати тільки команди:

1. Просунути вперед на 1.
2. Повернути направо на 90 градусів.
3. Повернути наліво на 90 градусів.

Автомат може також перевіряти умови:

1. Попереду стіна?
2. Вийшов з лабіринту?

Скласти алгоритм управління автоматом, що виводить його з лабіринту.

Задача 2.

(III обласна олімпіада, 1989 рік)

Дана прямокутна дошка розміром $M * N$ клітин. Частина клітин дошки заштриховані. Скласти алгоритм знаходження найкоротшого, що не містить заштрихованих клітин шляху з лівого верхнього в правий нижній кут дошки. Рух здійснюється з клітини в клітину по вертикалях і горизонталях.

Задача 3.

(III обласна олімпіада, 1989 рік)

Дана впорядкована за зростанням лінійна таблиця натуральних чисел A
 $[1] < A [2] < \dots < A [n]$. Знайти найменше натуральне число, не представляється у вигляді суми деяких чисел із цієї таблиці.

Пояснення. Сума може складатися і з одного доданка, але кожен елемент може входити в неї не більше одного разу.

Задача 4. «Рюкзак»

(VII обласна олімпіада, 1993 рік)

Із заданих n предметів вибрати такі, щоб їх сумарна маса була менша 30 кг, а вартість – найбільшою. Надрукувати сумарну вартість вибраних предметів.

Іншими словами – задано дві таблиці додатних чисел $A[1:n]$ і $B[1:n]$. Вибрати такі попарно різні числа i_1, i_2, \dots, i_k , щоб сума $A[i_1]+A[i_2]+\dots+A[i_k]<30$, а сума $B[i_1]+B[i_2]+\dots+B[i_k]=max$ була максимальною. Вивести лише величину *max*.

Зауваження: Можна вважати, що предмети вже розміщені в порядку зростання або спадання маси $A[i]$, або вартості $B[i]$, ціни $B[i]/A[i]$ чи якоїсь іншої ознаки.

Задача 5. «Перевезення вантажів»

(VII обласна олімпіада, 1993 рік)

Потрібно перевезти n ящиків, маса кожного з яких відома, найменшим числом рейсів на автомобілі, який вміщує m ящиків загальною масою не більше p кг. (n, m, p – задані натуральні числа).

Задача 6. «Бартер»

(VIII обласна олімпіада, 1994 рік)

В умовах інфляції багато підприємців віддають перевагу обміну своєї продукції на необхідні товари (бартеру), а не продажу продукції. Припустимо, що кожне підприємство може обміняти деяку кількість своєї продукції та визначає її ціну. Для кожного підприємства визначено множину підприємств, в продукції яких воно зацікавлене.

Підприємство може передати іншим будь-яку кількість свого товару в межах наявного, але не може пропонувати чужу продукцію. Біржа може виступати посередником в бартері між виробниками товарів, організуючи не тільки попарні обміни, а й більш складні, з дотриманням єдиної умови: кожен клієнт повинен отримати необхідні йому товари з такою ж сумарною вартістю, на яку він надав продукції іншим, наприклад: підприємство А надає підприємству В товару на 90 млн, В надає С та D товару відповідно на 70 та 20 млн, а С та D надають підприємству А товари на такі ж суми.

ЗАВДАННЯ: Написати програму, що для заданих пропозицій та запитів товарів запропонує таку комбінацію бартерних угод, щоб сумарна вартість обмінених була якомога більшою.

Задача 7. «Сходишки»

(IX обласна олімпіада, 1995 рік)

Людина піднімається по сходишках, або ступаючи на наступну сходишку, або перестрибуючи через одну чи дві сходишки. Знайти, скількома способами вона зможе піднятися на N -у сходишку ($N < 30$).

Задача 8. «Чотиризначні числа»

(IX обласна олімпіада, 1995 рік)

Знайти всі чотиризначні числа, кожне з яких записано різними цифрами і має наступні властивості: якщо цифри шуканого числа деяким чином переставити місцями і одержане таким способом нове чотиризначне число відняти від шуканого, то різницею буде чотиризначне число, записане тими ж цифрами. Вказати кількість таких чисел.

Задача 9. «Максимальна сума»

(IX обласна олімпіада, 1995 рік)

В клітини квадрата розміром $N \times N$ вписані довільно $N \times N$ перших чисел натурального ряду ($N \leq 7$). Від лівої верхньої клітинки до правої нижньої

прокладаються маршрути, причому клітинка старту і клітинка фінішу включаються в них, а дозволеними напрямками руху на одну клітинку вважаються два: вниз і вправо. Знайти кількість таких маршрутів. Організувати пошук маршруту, для якого сума чисел, записаних в клітинки, які йому належать, максимальна, і вказати цю суму. Передбачити введення чисел з клавіатури в клітинки квадрата, який відображається на екрані.

Задача 10. «Пірати»

(IX обласна олімпіада, 1995 рік)

N піратів знайшли скарб із золотих монет ($N \leq 10$). Один з них взяв собі одну монету і ще $1/N$ частину від тих монет, що залишилися. Точно так же зробили всі інші пірати. Монети, що залишилися, вони змогли поділити порівну. Знайти найменшу кількість монет, яка задовольняє описаному алгоритму поділу.

Задача 11. «Шляхи»

(XI обласна олімпіада, 1997 рік)

В одній з клітинок сторінки зошита знаходиться фішка. Горизонталь, на якій ця фішка міститься спочатку, будемо називати стартовою. Фішку можна пересувати ліворуч, праворуч, вгору або вниз. За один хід фішку можна пересунути на одну клітинку.

Дано: кількість ходів n , які дозволяється зробити.

Отримати:

- а) кількість шляхів, що повертають фішку на стартову горизонталь;
- б) перелік всіх шляхів, що повертають фішку на стартову горизонталь; кожен шлях представити у вигляді послідовності ходів (L - ліворуч, R - праворуч, U - вгору, D - вниз);
- в) дати графічну ілюстрацію шляхів фішки, що повертають її на стартову горизонталь; кожен наступний шлях має з'являтися на екрані при натисненні на будь-яку клавішу;

г) доступ до завдань а), б), в) здійснити через меню.

Задача 12. «Роман»

(XII обласна олімпіада, 1998 рік)

В романі N глав. В i -й главі a_i сторінок, $i = 1, \dots, N$. Потрібно видати роман в K томах так, щоб обсяг найбільшого тому був якомога меншим. Ділити главу між томами та переставляти порядок глав не можна. Напишіть програму NOVEL.*, що визначає оптимальний обсяг найбільшого тому.

Задача 13. «Завантаження автомобіля»

(XII обласна олімпіада, 1998 рік)

Є певний набір предметів p_1, p_2, \dots, p_N (кожний в одному примірнику); відомі їх маси q_1, q_2, \dots, q_N і вартості v_1, v_2, \dots, v_N . Вантажопідйомність автомобіля дорівнює Q . Визначіть, які предмети потрібно взяти на автомобіль, щоб їх сумарна вартість була максимальна.

Задача 14. «Паралелепіпед»

(XIII обласна олімпіада, 1999 рік)

Прямокутний паралелепіпед з натуральними довжинами сторін A, B і C (всі розміри не більші 10) розбито на кубики з ребром довжиною 1. Кубики пронумеровано випадковим чином послідовними натуральними числами від 1 до $A \cdot B \cdot C$. Через скільки кубиків пройде найкоротший шлях від кубика з номером 1 до кубика з номером $A \cdot B \cdot C$, якщо на кожному кроці можна переходити в будь-який сусідній (по вертикалі, по горизонталі, по діагоналі) кубик.

Задача 15. «Кубик»

(XIV обласна олімпіада, 2000 рік)

Дано кубик на гранях якого нанесені певним чином числа від 1 до 6 і прямокутну таблицю клітинки якої мають квадратну форму і площу рівну площі грані кубика. Клітинки таблиці заповнені натуральними числами 1 2,

3, 4, 5, 6 певним чином. Кубик ставлять у північно-західну клітинку таблиці. При цьому співпадання числа що написано на нижній грані кубика і числа у клітинці не обов'язково. Напишіть програму KUB *, (де * означає BAS, PAS чи CPP в залежності від мови програмування), яка визначає чи можна перемістити кубик у південно-східну клітинку таблиці, перекотивши його певну кількість разів через ребро слідкуючи при цьому, щоб число в клітинці таблиці і на нижній грані кубика співпадали. Якщо можна, то знайдіть хоча б один спосіб такого переміщення.

Вхідні дані

У першому рядку текстового ASCII-файла KUB.DAT через пропуск записані 6 чисел, які нанесені відповідно на північній, південній, східній, західній, верхній та нижній гранях кубика. У другому рядку через пропуск записані два натуральних числа M і N - висота (з півночі на південь) і ширина (із заходу на схід) таблиці. У кожному з наступних M рядків записані через пропуск по N чисел, які утворюють таблицю.

Вихідні дані

У першому рядку текстового ASCII-файла KUB.SOL потрібно записати NO або YES. Якщо переміщення здійснити можна, то в другому рядку записується число K - кількість перекочувань, в третьому рядку через пропуск записується K символів, які вказують спосіб перекочування. Для кодування переміщення слід використати такі позначення напрямів перекочування: N на північ, O - на схід, S на південь, W на захід.

Приклад вхідних і вихідних даних

KUB.DAT	KUB.SOL
312465	YES
58	11
52645264	OSOOSOOS
61634521	
12545262	
51635163	
13526454	

Задача 16. «Фірма»

(XV обласна олімпіада, 2001 рік)

З м.Кам'янця-Подільського до М.Славути вирушили дві супервантажівки за виробами на фірму "Олімпієць". Оплата праці водіїв пропорційна вантажу, який водії перевозять своїми вантажівками. Відомо, що на фірмі "Олімпієць" виготовлено N виробів, вага кожного з яких A_1, A_2, \dots, A_N .

Потрібно скласти програму FIRMA.PAS (FIRMA.BAS, FIRMA.CPP) завантаження обох вантажівок, щоб загальні маси вантажів обох вантажівок були максимально близькими. Вантажопідйомність вантажівок дозволяє забрати виготовлену продукцію. Ваша програма повинна прочитати дані з вхідного файлу FIRMA.DAT, а результат вивести у вихідний файл FIRMA.SOL

Вхідні дані:

У першому рядку текстового файлу FIRMA.DAT записане натуральне число N - кількість виробів; у наступних N рядках записані дійсні числа - маса i -го виробу в центнерах.

Вихідні дані:

У єдиному рядку текстового файлу FIRMA.SOL мають бути записані через пропуск загальні маси виробів, які завантажені на першу та другу вантажівки.

Приклад вхідних і вихідних даних.

FIRMA.DAT.

4

3.15

6.56

4.12

5.6

FIRMA.SOL

9.71 9.72

Задача 17. «Листки паперу»

(XV обласна олімпіада, 2001 рік)

Прямокутні листки на поверхні квадратного столу (розміром 100x100) розташовані так, що їх сторони лежать паралельно сторонам столу. Листки можуть перекриватись, але немає повністю закритих листків і таких, що дотикаються сторонами. Написати програму, яка визначає мінімально можливу кількість листків паперу, що забезпечують розташування згідно заданих координат видимих кутів листків.

Вхідні дані.

У першому рядку вхідного потоку задано кількість видимих кутів (≤ 20). У кожному наступному рядку (через пропуск) - пара координат, що визначає один видимий кут (X та Y).

Результат.

Вивести у вихідний потік кількість листків.

Задача 18. «Робінзон Крузо»

(XV обласна олімпіада, 2001 рік)

Після того як Робінзон Крузо побудував свого човна, він вирішив дослідити найближчі острови архіпелага з N островів. Метою своєї подорожі він вибрав N-ий острів, сподіваючись, що знайде на ньому туземне поселення і британського консула. Остерігаючись тропічних штормів, він вирішив не відпливати далеко від берега і перепливати з одного острова на інший, тільки якщо відстань між берегами цих островів (і, відповідно, плавання вдалині від берега) не перевищує D кілометрів. Для спрощення задачі всі острови будемо рахувати кругами з різним радіусом. Напишіть програму KRUSO.PAS (BAS, CPP), яка допоможе

Робінзону визначити, чи зможе він добратися до N-го острова. У вхідному текстовому файлі KRUSO.DAT в першому рядку записані два цілих числа N ($2 \leq N \leq 100$) і D ($5 \leq D \leq 50$), розділених одним пропуском - число островів і максимальна відстань, на яку відважиться здійснити плавання

Робінзон. В наступних N рядках знаходиться по три цілих числа X_i , Y_i , R_i ($0 \leq X_i$, $Y_i \leq 1000$, $K=R_i \leq 50$, $1 \leq i \leq N$) через один пропуск - координати і радіус острова. Перший острів є стартовим, а N-ий - метою подорожі.

У вихідний текстовий файл KRUSO.SOL вивести YES, якщо Робінзон зуміє добратися до мети, або NO, якщо мети неможливо досягнути для заданого обмеження D.

Приклад вхідних і вихідних даних.

KRUSO.DAT

59

5 25 3

8 12 2

15 15 5

30 15 3

35 5 4

KRUSO.SOL

YES

Задача 19. «Стиснення зображень»

(XVI обласна олімпіада, 2002 рік)

Чорно-біле зображення, що складається лише з білих та чорних точок, стискається таким чином. Якщо зображення складається лише з чорних точок, то в вихідний потік виводиться два біта - 0 та 0. Якщо зображення складається лише з білих точок, то в вихідний потік виводиться два біта - 0 та 1. Якщо ж зображення містить чорні та білі точки, то в вихідний потік виводиться біт 1, зображення розбивається на 4 частини, та кожна частина стискається цим самим алгоритмом окремо, спочатку перша, потім друга, третя та четверта:

1 2

4 3

Вихідний потік представляє собою стиснене зображення.

Наприклад, зображення

1	1	1	1	0	0	1	1
1	1	1	1	0	0	1	1
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

буде стиснене як 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0

Вхідні дані:

Файл test.img містить стиснене зображення розміром 256 на 256 точок.

Першими йдуть молодші біти байту.

Напишіть програму, що виводить зображення на екран в графічному режимі.

Задача 20. «Лабіринт»

(XVII обласна олімпіада, 2003 рік)

Маленький Паскаль заблукав у магічній печері, і ніяк не може знайти з неї вихід. Але маючи план печери-лабіринту та знаючи своє місцезнаходження, він вирішив написати на своєму кишеньковому комп'ютері програму, що визначить найменший час, за який він може дістатися виходу. План має вигляд квадрату розміром $N \times N$ ($3 \leq N \leq 50$), що розбитий на клітинки 1×1 . На плані лабіринту є такі умовні позначення:

"0" - вільна клітинка

"-1" - стіна

"-2" - клітинка де знаходиться Паскаль

"-3" - клітинка вихід

"A" - $1 \leq A \leq 10$ вхід до магічного телепорту з номером A

"B" - $101 \leq B \leq 110$ вихід з магічного телепорту з номером (B-100)

На кожному кроці хлопчина може переміститися на будь-яку сусідню по вертикалі чи горизонталі клітинку, що не є стіною. Якщо Паскаль потрапляє на клітинку, що містить вхід до телепорту, то він може (якщо захоче) переміститися до відповідного виходу (переміщення у телепорті відбувається миттєво, всі інші переміщення займають рівно одну хвилину). Для кожного магічного телепорту обов'язково існує вихід.

У першому рядку файлу MAZE.DAT знаходиться число N.

У наступних N рядках - план лабіринту.

У файл MAZE.SOL необхідно записати єдине число - час (у хвилинах), за який Паскаль дістанеться виходу з лабіринту. Ваша програма повинна мати назву MAZE.PAS (BAS чи CPP).

Приклад:

MAZE.DAT

5

0 0 0 0 0

0 7 0 -1 0

0 0 0 -1 -3

-2 0 0 -1 0

0 0 0 107 0

MAZE.SOL

6

Задача 21. «Копперфільд»

(XVIII обласна олімпіада, 2004 рік)

Кожний рік відомий ілюзійоніст Копперфільд проводить копперфільдівську лотерею. Переможець лотереї отримує право зіграти з

Копперфільдом в гру в його замку, результат якої визначає виграш переможця. Замок складається з $M \cdot N$ червоних і зелених квадратних кімнат, які розміщуються в шаховому порядку. В деяких кімнатах замку ввімкнене світло. Переможець лотереї (гравець) спочатку вибирає любую кімнату кольору, який вкаже Копперфільд, і в якій горить світло. В кожній кімнаті лежить чек на \$1000, і гравець, проходячи через кімнату, забирає чек собі. Щоб обмежити переміщення гравця, Копперфільд вимикає світло в деяких кімнатах, після чого гравець переходить в одну із сусідніх кімнат, в якій ще горить світло. Гра закінчується, коли гравець опиняється в єдиній світлій кімнаті. Тоді гравець виграє ту суму грошей, яку він зібрав, проходячи через кімнати замку. Якщо ж в процесі гри Копперфільд виключить світло в кімнаті, в якій знаходиться гравець, то гравець отримує всі $M \cdot N \cdot 1000$ доларів. Копперфільд діє так, щоб навіть при найбільш несприятливих для Копперфільда діях гравця, той зібрав якомога менше чеків. Визначити гарантований виграш гравця.

Пояснення.

На початку гри Копперфільд вибирає колір кімнати (або червоний, або зелений), а гравець займає любую кімнату вказаного кольору, в якій горить світло, причому Копперфільд не знає, яку саме. Гра проходить по кроках, причому і Копперфільд і гравець зобов'язані робити свої ходи. На кожному кроці спочатку Копперфільд виключає світло в довільних кімнатах по його вибору, потім гравець переходить в одну із сусідніх чотирьох кімнат, в якій ще горить світло. Гра закінчується при настанні однієї з таких трьох ситуацій:

Після ходу Копперфільда гравець опиняється в єдиній освітленій кімнаті в замку. В цьому випадку гравець забирає собі всі чеки, зібрані в кімнатах, через які він проходив на шляху до останньої кімнати;

Копперфільд виключає світло в кімнаті, в якій в даний момент часу знаходиться гравець. В цьому випадку Копперфільд програє всі чеки, розкладені у всіх кімнатах замку;

11110

11110

Задача 22. «Дід Мороз у замку»

(XIX обласна олімпіада, 2005 рік)

Стародавній замок має форму квадрата і містить $N \times N$ кімнат. У кожній кімнаті розставлені скриньки з золотими монетами. Дід Мороз знаходиться у верхній лівій кімнаті і мріє, зібравши якомога більше монет, добратися до правої нижньої кімнати, де його чекає чарівна Снігурочка. З кожної кімнати він може перейти до сусідньої внизу або сусідньої праворуч. Дід Мороз також хоче запам'ятати маршрут, який він пройде. Допоможіть Діду Морозу.

Формат вхідних даних: В першому рядку файла SANTA.CLS знаходиться одне число N ($1 \leq N \leq 30$). В кожному i -му з наступних N рядків знаходиться N чисел, що визначають кількість монет у кімнаті (i,j) .

Приклад введення:

5

7 1 4 9 3

3 8 4 2 1

8 1 6 7 7

2 7 4 4 5

4 5 2 6 5

Формат вихідних даних: Ваша програма повинна виводити у перший рядок файла SNOW.MDN одне ціле число – кількість монет, яку вдалося зібрати Діду Морозу. У другому рядку – маршрут – номери кімнат, у яких побував Дід Мороз, починаючи з кімнати $(1,1)$, і закінчуючи (N,N) .

Приклад виведення:

52

(1,1) (2,1) (2,2) (2,3) (3,3) (3,4) (3,5) (4,5) (5,5)

Задача 23. «Chess»

(XX обласна олімпіада, 2006 рік)

Піхотинець проти вершника! Ніяких шансів для перемоги!? Подивимось. Отже, поле бою – шахова дошка розмірності $N \times N$ ($4 \leq N \leq 100$). На дошці знаходяться кінь та пішак. Ваша задача написати програму, що визначає кількість ходів, які потрібно зробити коневі, щоб побити пішака, який прямує у дамки, або вивести 0, якщо цього він зробити не зможе. Пішак ходить лише на одну клітинку вперед; якщо пішак дійшов у дамки (на перший рядок таблиці) і кінь має черговий хід у цю клітинку, то мрії пішака не здійснилися.

Вхідний файл **chess.in** містить єдиний рядок у якому знаходяться в такому порядку цілі числа через пропуск: перше число N – розмір дошки, друге – 0 або 1 (1 – перший хід пішака, 0 – перший ходить кінь), далі записані координати коня та пішака.

Вихідний файл **chess.out** містить єдине число – кількість ходів коня, або 0.

Задача 24. «Подарунок для Мудрої Сови»

(XX обласна олімпіада, 2006 рік)

П'ятачок і Вінні-Пух зібрались до Мудрої Сови на день народження. В якості подарунка вирішили назбирати букет польових ромашок. Ромашкове поле задано прямокутною матрицею $M \times N$, в клітинках якої записано кількість ромашок, що ростуть на даному клаптику поля. Будь-який клаптик поля містить хоча б одну ромашку.

П'ятачок і Вінні-Пух вміють ходити лише по горизонталі і вертикалі, при цьому, зробивши хід на схід, уже не можна буде потім ходити на захід і навпаки, а також, зробивши хід на північ не можна буде ходити на південь і навпаки. В початковий момент гості Мудрої Сови знаходяться в центральній клітинці поля.

Напишіть програму, що допоможе вказати напрямок руху: **NW** – північний захід, **NE** – північний схід, **SW**- південний захід, **SE** – південний схід. Та визначте при цьому максимально можливу кількість ромашок у букеті.

Технічні умови:

У першому рядку вхідного текстового файлу **owl.dat** записано через пропуск два цілих непарних числа: M і N . ($2 < M, N < 100$).

В наступних M рядках записано по N чисел у кожному розділених пропуском. Кожне з чисел не перевищує 10000.

Ваша програма повинна вивести у файл **owl.sol** напрямок руху у вигляді: **NW** – північний захід, **NE** – північний схід, **SW**- південний захід, **SE** – південний схід. Та через пропуск у відповідному рядку кількість ромашок. Якщо таких напрямків буде більше одного, то вивести всі від **NW** за годинниковою стрілкою.

Приклад 1.

owl.dat	owl.sol
3 1 2 3 2 1 1 1	NE 7

Приклад 2.

owl.dat	owl.sol
3 1 2 1 2 3 2 1 2 1 1 3 1 1	NW 7 NE 7 SE 7 SW 7

Задача 25. «Знайомства»

(XX обласна олімпіада, 2006 рік)

На олімпіаду з інформатики приїхало N учасників ($1 \leq N \leq 200$). Деякі з них вже знайомі між собою. Вам необхідно написати програму, яка визначає,

скільки вийде груп знайомих, після того, як учасники перезнайомляться між собою опосередковано (через спільного знайомого).

Вхідні дані:

У вхідному потоці в першому рядку міститься число N . Далі, в N рядках задані цілі числа, розділені пропусками, які позначають номери учасників, з якими знайомий цей учень. Якщо учасник ні з ким не знайомий, то в рядку міститься значення 0.

Вихідні дані:

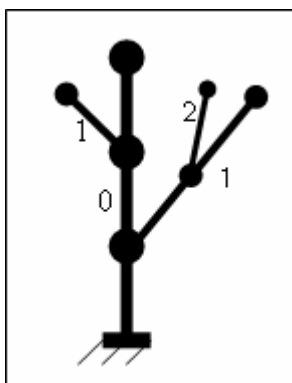
У вихідний потік необхідно вивести одне число – кількість груп.

Наприклад:

Input	Output
6	3
5	3
4	
1	
2	
1	
0	

Задача 26. «Золоте дерево»

(XX обласна олімпіада, 2006 рік)



Як відомо, дурнувятий Буратіно за порадою кота Базілію і лисиці Аліси рівно опівночі на Полі чудес в Країні дурнів посадив «золотий». В перший день з «золотого» виросте стовбур одиничної довжини. За кожний наступний день кожна гілка чарівного дерева, включаючи стовбур, виростає в тому ж напрямі ще на одну одиницю довжини. При цьому, в кінці кожної гілки зав'язується одна нова брунька. Наступного дня із зав'язаної бруньки виростають нові гілки одиничної довжини і т.д. По законах казкової ботаніки в повний місяць на гілках, які не молодше K -го і не старше за N -го порядку, з ще не пророслих бруньок замість паростків з'являються золоті плоди. (Порядок гілки

визначається її положенням відносно стовбура: стовбур має нульовий порядок, гілки, що ростуть від стовбура – перший порядок, від гілок першого порядку – другий і т.д., див. рис).

Вам необхідно написати програму, яка дозволяє визначити кількість золотих плодів, якщо повний місяць відбудеться через T днів ($1 \leq T \leq 28$).

Вхідні дані:

Вхідний потік містить три цілі числа T , K і N , розділені між собою пропусками.

Вихідні дані:

У вихідний потік необхідно вивести одне число – кількість золотих плодів.

Наприклад:

Input	Output
3 1 3	3

Задача 27. «Музична колекція»

(XX обласна олімпіада, 2006 рік)

Школяр Вася зібрав музичну колекцію, яка складається з M улюблених пісень. Щоб слухати пісні, Вася вирішив записати їх на N дисків, кожен з яких може вміщувати не більш ніж P хвилин музики за такими правилами:

1) пісні повинні розташовуватися в порядку зростання номерів, тобто для будь-яких двох дисків усі пісні з одного повинні мати менші номери, ніж пісні з другого;

2) кількість записаних пісень повинна бути максимально можливою.

Знаючи скільки хвилин займає кожна пісня, допоможіть Васі впоратися із поставленим завданням.

Вхідні дані.

В першому рядку файлу **music.dat** знаходяться числа N , P , M розділені пропуском. ($1 \leq N, P, M \leq 30$).

В наступних M рядках по одному числу: в i -му рядку розмір у хвилинах пісні з номером $(i-1)$. Всі вхідні дані - цілі числа.

Вихідні дані.

В перший рядок файлу **music.sol** вивести максимальну кількість пісень, яку Вася зможе записати на диски.

Приклад вхідних і вихідних даних.

music.dat	music.sol
2 5 3	2
2	
4	
3	

Задача 28. «Глобальне потепління»

(XX обласна олімпіада, 2006 рік)

Кому з вас не відомий знаменитий мультфільм режисерів Кріса Уейджа та Карлоса Сайдана «Льодовиковий період»? Говорять, що після «Льодовикового період-2» вийде «Льодовиковий період-3» і в перспективі почнуться роботи над створенням нового катастрофічного мультфільму «Глобальне потепління-1».

Ізюминку сюжету цього фільму становить момент, коли ненаситна білочка Скрет попала в Долину Жолудів. Крім величезних харчових запасів, Скрет вразила сама місцевість. Вся долина – це певна кількість горбів, що являють собою півсфери різних радіусів. Звісно, ці симпатичні горбики рясніють від великої кількості жолудів. Все було б для Скрет чудово, якби Людина своєю техногенною діяльністю не спровокувала глобальне потепління. Чергова катастрофа застала білочку посередині цієї широкої долини. Долину почало швидко затоплювати і Скрет в паніці стрибала з одного горбика на інший, шукаючи порятунку. Білка, завдяки великим запасам жолудів мала достатньо енергії щоб стрибати на будь який горбик, що знаходився не далше, ніж L метрів від горбика, на якому знаходилася

вона сама. Відстань між горбиками – це відстань між центрами півсфер незалежно від їх радіуса. За кожну хвилину вода піднімалася на висоту H метрів. Отже, висота видимої з води частини півсфери також зменшувалися на H метрів за кожну хвилину. Скрет бачила, що один за одним зникають під водою рятівні острівки землі. Для того щоб врятуватися, вона повинна швидко добратися найбільшого горбика-острівка. Допоможіть Скрет знайти рятівний шлях до найбільшого острівка, якщо вона за 1 хвилину може здійснити лише один стрибок. Білочка не може стрибати на острівцець, що зрівнявся з поверхнею води, але завжди встигає покинути острівцець, який має бути затоплений на протязі наступної хвилини.

Формат вхідних даних: У першому рядку вхідного файлу **skret.dat** знаходяться три цілих числа N, L, H ($2 \leq N \leq 100, 0 < L \leq 1000, 0 \leq H < 100$). У наступних N рядках дано по три цілих числа X_i, Y_i, R_i ($-10000 \leq X_i, Y_i \leq 10000, H < R_i \leq 10000$). Скрет у початковий момент часу знаходиться на першому острові. Нижні частини острівців можуть перекриватися, але ніякі два центри не співпадають.

Формат вихідних даних: У випадку наявності рятівного шляху до найбільшого острова у перший рядок вихідного файлу **skret.sol** вивести найменшу кількість стрибків Скрет, а у другий рядок через пропуск вивести номери островів, на які вона повинна послідовно стрибати. Якщо є декілька найбільших островів – вивести шлях до острова з мінімальним номером. У випадку відсутності рятівного шляху у єдиний рядок вихідного файлу вивести повідомлення «**The life is fine**» без лапок.

Приклад вхідних та вихідних даних:

skret.dat

6 4 1

1 1 1

0 3 2

3 5 1

6 6 4
4 0 2
6 2 3
skret.sol
3
5 6 4
skret.dat
2 10 10
0 0 5
10 10 20
skret.sol
The life is fine

Задача 29. «Їжачок»

(XXII обласна олімпіада, 2008 рік)

На шахівниці розміром $N \times N$ розставлено K пронумерованих фішок. Їжачок, який збирає ці фішки, виходить з клітки з координатами $(1,1)$ і повинен зібрати всі фішки в порядку зростання їх номерів. З клітки з координатами (x,y) їжачок може переміститися тільки в одну з чотирьох сусідніх, тобто в одну з кліток з координатами $(x+1,y)$, $(x-1,y)$, $(x,y+1)$ або $(x,y-1)$, звичайно з дотриманням умови, що він не повинен виходити за межі дошки. Вимагається визначити, яку мінімальну кількість ходів потрібно зробити, щоб зібрати всі фішки. Якщо їжачок проходить через клітку, де міститься фішка з більшим номером, ніж він зараз повинен узяти, то він просто проходить, а фішка залишається стояти.

Вхідні дані. Ваша програма повинна вводити з клавіатури спочатку два числа розділених пропуском N і K ($1 \leq N, K \leq 1000$). Далі у K рядках по два цілі числа, розділені пропуском – координати фішок в порядку зростання їх номерів, тобто спочатку для першої фішки, потім для другої фішки і т.д. ($1 \leq x, y \leq N$).

Вихідні дані. Ваша програма повинна вивести на екран одне число – мінімальну кількість кроків їжачка, необхідних для збирання всіх фішок.

Формат вихідних даних:

У вихідний файл необхідно вивести

Приклади вхідних і вихідних даних:

Вхідні дані:	Вихідні дані:
4 3	11
3 3	
1 4	
2 1	

Задача 30.

(XXIII обласна олімпіада, 2009 рік)

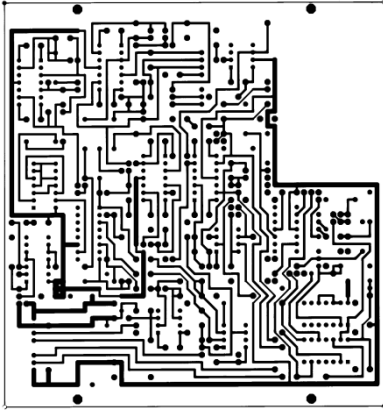
Існує деяка вулиця з одностороннім рухом, що має M місць для паркування автомобіля з номерами від 1 до M . В автомобілі, що їде цією вулицею, рядом з чоловіком-водієм дрімає його дружина, яка раптом просипається і вимагає негайно зупинитися. Чоловік слухняно виконує бажання жінки і зупиняється на першому вільному місці для паркування, що знаходиться попереду. Якщо ж вільних місць до кінця вулиці немає, то законслухняність водія перевищує слухняність жінці і він, вибачаючись, їде далі.

Нехай, на цій вулиці опинилися N автомобілів із жінками, що дрімають рядом з водієм. Причому, жінка з j -го автомобіля проснеться навпроти $a[j]$ -го місця паркування. Знайдіть одну з можливих послідовностей паркування автомобілів або максимальну кількість припаркованих автомобілів у випадку, коли не всі автомобілі зможуть припаркуватися. Будемо вважати, що всі місця паркування на початку були вільні і жоден із автомобілів не покидає своє місце паркування.

Задача 31.

(XXIII обласна олімпіада, 2009 рік)

Гуртківці Славутської СЮТ вирішили сконструювати монтажну плату з мінімальною індуктивністю провідних доріжок.



Як відомо, прямий провідник має меншу індуктивність, ніж провідник, що містить повороти. Отже, юним радіотехнікам треба навчитися розв'язувати задачу: як з'єднати точку А із точкою В доріжкою з мінімальною кількістю поворотів. Давайте допоможемо гуртківцям у знаходженні такого шляху від точки А до точки В.

Отже, маємо прямокутну плату $N \times M$ клітинок. Кожна з клітин або зайнята провідником, або вільна. Доріжки можна будувати лише паралельно сторонам плати, повороти провідника можна робити лише в середині клітин. Кожна клітина може містити лише одну провідну доріжку. Знайдіть шлях від клітини А до клітини В з мінімальною кількістю поворотів.

У першому рядку вхідного файлу `input.txt` містяться цілі числа $N, M (0 < N, M < 1000)$. У наступних N рядках містить по M символів: `.` – порожня клітинка, `*` – клітинка зайнята провідником, `A` – початкова точка, `B` – кінцева точка.

У першому рядку вихідного файлу `output.txt` вивести мінімальну кількість поворотів на шляху від точки А до точки В, а в наступних – один із можливих шляхів у форматі наведеного прикладу. Якщо шляху не існує, то вивести «There is no way».

Задача 32. «Lines»

(XXIV обласна олімпіада, 2010 рік)

У таблиці із N рядків і M стовпців деякі клітинки зайняті шариками, інші вільні. Вибрали шарик, який треба перемістити, і місце, куди його треба перемістити. Вибраний шарик за один крок переміщується в сусідню по

горизонталі або вертикалі вільну клітинку. Потрібно в'яснити, чи можливо перемістити шарик із початкової клітинки у задану, і якщо можливо, то знайти шлях із найменшою кількістю кроків.

Задача 33. «Подорож м'ячика по сходинках»

(Заочна олімпіада, 2002-2003)

На вершині драбинки, що складається з N сходинок, знаходиться м'ячик, який починає котитися, підскакуючи, донизу. М'ячик під час своєї "подорожі" може скочити на наступну сходинку, через одну або через дві сходинки. (Тобто, якщо м'ячик знаходиться на 10 сходинці, то він може переміститися на 9-у, 8-у і 7-у сходинки). Визначити число всіх можливих маршрутів "подорожі" м'ячика на землю.

Технічні умови.

У файлі BALL.DAT міститься єдине число $0 < N < 31$.

У вихідному файлі BALL.SOL міститься одне число - кількість маршрутів.

Задача 34. «НЕП»

(Заочна олімпіада, 2002-2003)

У деякій фірмі, не обтяженій засобами виробництва, для випуску N видів готової продукції працює дві бригади. Для j -го виробу першій бригаді необхідний час $a[j]$, потім час $b[j]$ потрібен 2-й бригаді для упаковки виробу. В якій послідовності необхідно виготовляти вироби, щоб загальний час для виготовлення і упаковки всієї продукції був мінімальним. Всі види продукції обробляються бригадами в одній послідовності.

Технічні умови:

Вхідний файл NEP.DAT у першому рядку містить одне число $0 < N < 31$ - кількість видів продукції. Другий - N цілих чисел - часові затрати першої бригади для випуску кожного виду продукції. Третій рядок - N цілих чисел - часові затрати другої бригади для упаковки кожного виду продукції.

Наприклад:

5

4 4 30 6 2

5 1 4 30 3

Вихідний файл NEP.SOL у пешому рядку містить N цілих чисел - оптимальний порядок випуску видів продукції, другий - одне ціле число - загальний час виготовлення всіх видів продукції.

Наприклад:

5 1 4 3 2

47

Задача 35. «Фабрика»

(Заочна олімпіада, вища ліга, 2005-2006)

На фабриці працює N робітників. Для переоснащення виробництва нещодавно завезли K нових однакових станків, на яких можна виготовляти надсучасні пристрої. Але виникла одна проблема: лише M робітників вміють працювати за цими станками. Власник фабрики наказав вмілим робітникам допомогти своїм колегам.

За один день кожен робітник, що вміє працювати за станками, може навчити рівно одного іншого робітника, але для цього потрібно зайняти на день окремий станок, бо вчитись можна лише на практиці. За кожним станком можна навчати лише одного робітника в день. Через скільки днів всі робітники будуть вміти виготовляти надсучасні пристрої при оптимальному плані навчання?

Технічні обмеження : $(1 \leq N \leq 10^9)$, $(1 \leq M < N)$, $(1 \leq K \leq 10000)$.

Вхідні дані. В першому рядку файла **plant.dat** записані числа N , K , M через пропуск.

Вихідні дані. В перший рядок файлу **plant.sol** вивести одне число - мінімальну кількість днів,

за які можна навчити всіх робітників.

Приклад вхідних і вихідних даних.

plant.dat

8 3 2

plant.sol

3

Задача 36. «Гірський туризм»

(Заочна олімпіада, вища ліга, 2005-2006)

Один завзятий турист вирішив влітку помандрувати Карпатськими горами і підкорити одну відому вершину. Після кількох днів неперервних мандрів турист витратив чимало сил, але так і не дістався місця призначення. Тепер йому потрібно економити сили і добратися до потрібної вершини з якомога меншим зусиллям.

Карта території, по якій можна переміщуватись, представлена прямокутною таблицею розміру $M \times N$. Кожна клітинка таблиці задає висоту відповідної вершини в метрах. Турист зараз знаходиться на вершині з координатами (x_1, y_1) , а вершина, на яку він хоче потрапити, має координати (x_2, y_2) . Турист витрачає по одній одиниці своєї сили на кожен метр підйому чи спуску. Переміщуватись він може лише між тими вершинами, яким на карті відповідають сусідні клітинки. Так із вершини з координатами (x, y) можна переміститись лише у вершини $(x-1, y)$, $(x+1, y)$, $(x, y-1)$, $(x, y+1)$, якщо вони знаходяться в межах карти.

Обрахуйте мінімальну кількість одиниць сили, яку турист повинен витратити, щоб дістатись потрібної вершини.

Вхідні дані. В першому рядку файлу **map.dat** записані числа M , N . ($2 \leq M, N \leq 100$). В другому рядку знаходяться числа x_1 , y_1 , x_2 , y_2 . Далі йдуть M рядків по N чисел від 0 до 2000 , які задають висоти вершин на карті. В i -му рядку на j -му місці записана висота вершини з координатами (i, j) .

Вихідні дані. В перший рядок файлу **map.sol** записати єдине число - мінімальну кількість одиниць сили, яку турист повинен затратити на подорож до потрібної вершини.

Приклад вхідних і вихідних даних.

map.dat

3 2

1 1 3 1

100 200

900 500

300 400

map.sol

600

Задача 37. «Б-дерево»

(Заочна олімпіада, вища ліга, 2005-2006)

Владислав дуже цікавиться програмуванням, а особливо різними структурами даних. Зараз він намагається опанувати Б-дерева. Б-Дерево – це дерево, кожна вершина якого має рівно одного предка (крім кореня, який їх немає зовсім) і n дітей. Тож Владислав намагався намалювати на папірчику Б-дерево. Щоб детальніше розібратися, Владислав намалював в корені число m . Потім для кожної вершини, в якій знаходилося деяке число більше одиниці, він домалював n дітей, причому кожне з них було цілим і їхня сума дорівнювала числу у вершині, яку він розглядав. Владислав також намагався, щоб різниця між максимальним і мінімальним намальованим числом була як можливо менша. В кінці дня, закінчивши цю складну і виснажливу роботу, Владислав намагався порахувати суму написаних чисел, але так як Б-дерево було досить великим, то він кожного разу збивався з рахунку, і в результаті він звернувся за допомогою до вас.

Формат вхідних даних: у єдиному рядку вхідного файлу міститься два числа n ($1 \leq n < 2^{31}$) і m ($2 \leq m < 2^{31}$).

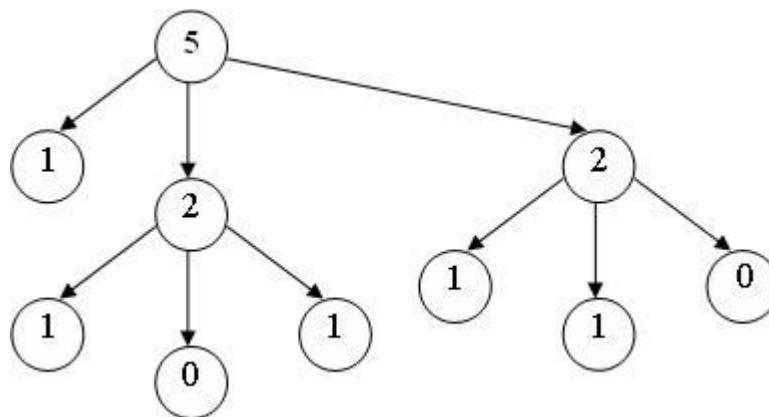
Формат вхідних даних: у єдиний рядок вихідного файлу виведіть єдине число – суму написаних Владиславом чисел.

Приклад введення і виведення:

btree.in	btree.out
3 5	14

Пояснення :

Нижче зображено, намальоване Владиславом дерево.



Задача 38. «Археологія»

(Заочна олімпіада, вища ліга, 2006-2007)

Археологи проводять розкопки старовинного міста. Вони встановили, що у різні періоди, на одному і тому ж місті були встановлені різні споруди, площа фундаментів яких можуть перетинатися. Вони створили "карту фундаментів" і хочуть взнати загальну площу, яка була покрита спорудами.

Відомо, що фундаменти усіх споруд мали форму прямокутника.

Вхідні дані (файл **FUND.DAT**):

У першому рядку знаходиться число **N** - кількість фундаментів. ($0 < N \leq 3000$).

У кожному з наступних **N** рядків знаходяться координати лівого верхнього і правого нижнього кутів кожного з фундаментів. (координати - цілі числа в межах від 0 до 1000000).

Вихідні дані (файл **FUND.SOL**):

У вихідний файл записати єдине число - площу покриту фундаментами.
Результат записати з трьома знаками після коми.

Приклад:

FUND.DAT	FUND.SOL
3	700.000
20 30 40 10	
10 40 30 20	
23 25 25 15	

Задача 39. «Проблема «множення»

(Заочна олімпіада, вища ліга, 2006-2007)

У гру «Множення» грають з рядом карт, кожна з яких містить одне додатне ціле число. За один хід гравець забирає одну карту з ряду і отримує певне число балів, рівне добутку числа на забраній карті і чисел на картах, що лежать безпосередньо зліва і справа від неї. Не дозволяється прибирати першу і останню карти ряду. Після останнього ходу у ряду залишаються тільки дві карти.

Мета гри - прибрати карти у такому порядку, щоб мінімізувати загальну кількість набраних балів.

Наприклад, якщо карти містять числа 10, 1, 50, 20 і 5, гравець може взяти карту з числом 1, потім 20 і 50, отримуючи бали

$$10 * 1 * 50 + 50 * 20 * 5 + 10 * 50 * 5 = 500 + 5000 + 2500 = 8000.$$

Якби би він взяв карти у зворотному порядку, тобто 50, потім 20, потім 1, кількість набраних балів була б такою:

$$1 * 50 * 20 + 1 * 20 * 5 + 10 * 1 * 5 = 1000 + 100 + 50 = 1150.$$

Формат вхідних даних: у першому рядку файлу **mpuzzle.in** знаходиться одне число – кількість карт **N**, у другому - розділені пропусками **N** чисел на картах.

Обмеження: $3 \leq N \leq 100$, числа на картах цілі від 1 до 100.

Формат вхідних даних: виведіть у файл **mpuzzle.out** одне ціле число - мінімально можливе число балів.

Приклад введення і виведення:

mpuzzle.in	mpuzzle.out
6	3650
10 1 50 50 20 5	

Задача 40. «Велика покупка»

(Заочна олімпіада, вища ліга, 2006-2007)

Школяр Вася Пупкін хоче купити собі новий портфель. Він не знає точної ціни, але впевнений, що вона не перевищує **K** грошових одиниць. В заощадженнях Васи є купюри **N** різних номіналів. Він не хоче нести з собою в магазин велику кількість купюр, але хоче, щоб за допомогою взятих купюр можна було без задачі купити портфель.

Вхідні дані: В першому рядку файлу **buying.dat** числа **K** і **N** ($1 \leq K \leq 1000$, $1 \leq N \leq 10$). В другому рядку **N** цілих чисел в межах від 1 до 1000 – доступні номінали купюр.

Вихідні дані: У вихідний файл **buying.sol** потрібно вивести мінімальну кількість купюр, які необхідно взяти, щоб можна було сплатити будь-яку суму від 1 до **K**, або -1 у випадку, якщо цього зробити не можливо. Купюру кожного номіналу можна брати довільну кількість разів.

Приклад вхідних і вихідних даних:

buying.dat

20 4

1 2 5 10

buying.sol

5

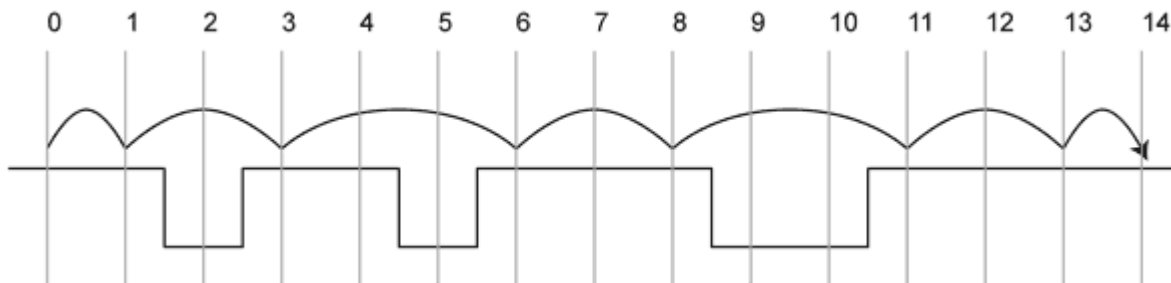
Задача 41. «Біг з перешкодами»

(Заочна олімпіада, вища ліга, 2006-2007)

Кенгуру знаходиться на старті бігової доріжки. Кенгуру може стрибати на відстань від 1 до 10 метрів. Але щоб стрибнути на 10 метрів йому потрібно добряче розігнатися. Завдяки явищу інерції кожен наступний стрибок може бути або на 1 метр довший або на 1 метр коротший за попередній, також може бути рівний попередньому. Відповідно кенгуру не може зупинитися в будь-який момент, а лише коли довжина попереднього стрибка була рівною 1 метру.

На біговій доріжці є ряд перешкод у вигляді ям різної ширини. Попавши в яму кенгуру більше не може з неї вилізти.

Будь-який стрибок кенгуру займає 1 секунду часу. За який найменший час кенгуру може фінішувати на заданій доріжці? Кенгуру повинен зупинитися точно на фініші. Кенгуру не повинен попадати в жодну яму. Якщо яма знаходиться на фініші, то вважати фініш недосяжним.



Початкові дані:

В першому рядку вхідного потоку міститься одне ціле число L – довжина бігової доріжки в метрах, $1 \leq L \leq 30000$. В другому рядку вхідного потоку міститься L чисел розділених пропусками, що позначають стан бігової доріжки в позиціях 1, 2, 3, ..., L метрів відповідно. "0" позначає добрий стан доріжки, "1" позначає яму в заданій позиції. Інших позначень немає.

Результат:

В перший рядок вихідного потоку вивести єдине ціле число – найменший час в секундах, за який кенгуру може сягнути фінішу за умови

повної зупинки на фініші. Якщо кенгуру взагалі не може досягнути фінішу, вивести "-1".

Приклад початкових даних:

Початкові дані, що відповідають малюнку наведеному вище:

14

01001000110000

Приклад результату:

Результат, який відповідає початковим даним, що наведені вище:

7

Задача 42. «Державна загроза»

(Заочна олімпіада, вища ліга, 2006-2007)

Колишній учасник олімпіади Петренко (прізвище змінене з етичних міркувань) ставши дуже багатою і впливовою людиною, захотів взяти під контроль законодавчий орган країни – Верховну Раду.

Для цього, він вирішив сформувати свою коаліцію, яка буде відстоювати його олігархічні інтереси. Менеджери Петренка зібрали інформацію по кожній фракції, що містить інформацію – кількість депутатів у фракції та сума, за яку можна підкупити цю фракцію.

Але жадібність Петренка хоче, аби сформувати більшість, і витратити при цьому якнайменше грошей.

Про це довідався полковник Іваненко (прізвище змінене з етичних міркувань), що працює в Службі Безпеки України. Розуміючи що над країною зависла загроза олігархічного тоталітаризму Іваненко звернувся до вас по допомогу. Напишіть програму, яка спрогнозує які фракції хоче підкупити Петренко.

Вхідні дані:

В першому рядку вхідного потоку міститься число N – кількість фракцій у Верховній Раді.

В кожному з наступних N рядків міститься по два числа P і K - кількість депутатів у фракції і сума коштів необхідна на підкуп фракції.

Примітка: $1 \leq N \leq 100$, $1 \leq K \leq 10000$. Загальна кількість депутатів парламенту не перевищує 10000. Більшість – будь-яка кількість депутатів, що більша від половини загальної кількості депутатів.

Вихідні дані:

В першому рядку два числа M – мінімальна сума, що необхідна для створення більшості у Верховній Раді, та V – кількість фракцій, які необхідно підкупити.

У наступному рядку міститься V цілих чисел – номери фракцій, які необхідно підкупити.

У випадку, коли розв'язків може бути декілька, вивести будь який з них.

Приклад вхідних даних:

5
27 1
175 500
20 44
72 200
156 450

Приклад вихідних даних:

650 2
4 5

Задача 43. «Виставка»

(Заочна олімпіада, 2009-2010)

Кожної осені Хмельницька обласна адміністрація проводить сільськогосподарську виставку, на яку приїздять поважні фермери, щоб себе показати і на людей подивитися. На таку виставку поїхали наші герої Дієтенко та Вампіров зі своїми кроликами. Їхній транспорт має обмежену

вантажопідйомність. Ось вони задумалися над тим, яку мінімальну та максимальну суми вони отримають, реалізувавши свій товар. Їм відома E - маса автомобіля не навантаженого, F – повна маса атомобіля, маса одного кролика M_i та його вартість P_i . Допоможіть їм знайти максимальну та мінімальну суми, які вони можуть отримати.

Вхідні дані: у текстовому файлі `market.in` у першому рядку записані числа E і F : $1 \leq E \leq F \leq 10000$. У другому рядку – число N : $1 \leq N \leq 500$ - кількість кроликів, а у наступних N рядках пари чисел M_i та P_i : $1 \leq P_i \leq 50000$, $1 \leq M_i \leq 10000$, які розділені пропуском. Всі числа натуральні.

Вихідні дані: у текстовий файл `market.out` записати через пропуск два числа: максимальну та мінімальну суми. Якщо автомобіль не може мати точно задану вагу при умові, що він навантажений заданими кроликами, то вивести «This is impossible.».

Приклад 1:

Вхідний файл: market.in

1000 1100

2

1 1

5 2

Вихідний файл: market.out

250 100

Приклад 2:

Вхідний файл: market.in

1000 2000

1

10 3

Вихідний файл: market.out

This is impossible.

Задача 44. «Обласна олімпіада»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Обласна олімпіада з інформатики, це не лише випробування для учасників. Кожного разу перед командою журі постає ціла низка важких завдань. Деякі з них відомі заздалегідь, деякі виникають в процесі олімпіади, і потребують негайного рішення. Команда журі складається з N людей, кожен з яких може виконувати певний набір функцій з різною результативністю. Наприклад KAV найкраще може скласти тести до задач, SSF - найкраще підходить на функцію голови експертної комісії. PSS - найкраще розбирається в комп'ютерних мережах і недопустить можливість доступу з комп'ютера одного учасника до іншого, тощо. Кожному члену журі можна призначити лише одну роботу. Найголовніша задача кожної олімпіади полягає в тому, щоб голова журі RVA розподілив обов'язки між членами журі найкращим чином.

Вхідні дані:

В першому рядку міститься ціле число N ($1 < n \leq 100$)- кількість членів журі. Далі йде $N * N$ чисел які описують матрицю робіт. Число A_{ij} ($1 \leq A_{ij} \leq 10000$) - показує, з якою ефективністю член журі під номером i виконає роботу під номером j .

Вихідні дані:

Виведіть єдине число - найбільшу сумарну ефективність, яку можна досягнути для заданої олімпіади

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
3 3 6 8 7 1 9 3 5 6	20

Довга арифметика

Задача 1. «Дроби»

(XIII обласна олімпіада, 1999 рік)

В десятковому запису дробу m/n викреслили k -ту цифру після коми. Порівняти (менше, більше, рівно) отримане число з дробом m/n .

Задача 2. «Квадратний корінь»

(XIV обласна олімпіада, 2000 рік)

Напишіть програму ROOT.*, яка обчислює корінь квадратний з числа n ($0 < n < 10^{2000}$) з точністю k ($k < 19$) десяткових знаків без використання стандартної функції обчислення квадратного кореня.

Вхідні дані

У текстовому файлі ROOT.DAT через пропуск записані два числа n і k .

Вихідні дані

У текстовому файлі ROOT.SOL потрібно записати єдине число, яке є коренем квадратним числа n з точністю k десяткових знаків.

Приклад вхідних і вихідних даних

ROOT.DAT	ROOT.SOL
64.23 3	8.014

Задача 3.

(Заочна олімпіада, 2000-2001)

Розглянемо всі натуральні числа від 1 до N . Скільки разів зустрічається знак 0 при записування всіх цих чисел.

Вхідні дані: в текстовому файлі ZEROES.DAT записане натуральне число N (число має не більше 1000 знаків).

Вихідні дані: в текстовий файл ZEROES.OUT виведіть кількість нулів.

Приклад вхідних даних №1:

999

Приклад вихідних даних №1:

189

Приклад вхідних даних №2:

123456

Приклад вихідних даних №2:

58985

Задача 4. «Crazy frog»

(Заочна олімпіада, вища ліга, 2005-2006)

Юні науковці досліджували поведінку нового виду жаб - "Крейзі Фрогс". Обмеживши зону стрибка жаби числовою прямою, використовуючи мікромметр, вони отримали неймовірно цікавий результат: кожна жаба робила два довільні (різні за довжинами) стрибки вліво, або вправо на числовій прямій, а третій - математично залежав від перших двох. Залежність була такою: якщо довжини перших стрибків відрізнялись більш ніж в двічі - "Крейзі Фрог" третій стрибок робила (в напрямку протилежному до сумарного зміщення) на довжину абсолютної різниці перших двох стрибків, а в іншому випадку - "Крейзі Фрог" робила "супер стрибок", числове значення якого дорівнювало добутку числових значень перших двох стрибків. Стрибки вправо вважалися додатніми, а вліво - від'ємними. Необхідно розробити програму, яка обчислює по першим двом стрибкам точне значення третього стрибка.

В двох рядках текстового файлу CRAZY.FRG знаходяться числові значення відповідних стрибків "Крейзі Фрог" (записані в метрах). Виведіть на екран числове значення третього стрибка.

Приклад 1 файлу CRAZY.FRG:

5

2.002

Приклад 1 вихідних даних:

-2.998

Приклад 2 файлу CRAZY.FRG:

-2

3

Приклад 2 вихідних даних:

-6

Задача 5. «Грошова реформа»

(Заочна олімпіада, вища ліга, 2006-2007)

Правитель двійкової планети Бінаріяндії президент Бінар вирішив зробити грошову реформу, тепер цінність кожної купюри достоїнством в P бінарів буде кількість одиничних бітів в двійковому записі числа P . Опозиціонер Дециміал, який для зміцнення зв'язків з планетою Земля вже давно пропонував ввести десяткову систему числення, а також змінити назву планети на Дециміаліяндію і ім'я президенту (на яке, досить не повідомляється) думає що це сильно вдарить по його капіталу. Він має по одній купюрі достоїнства $L, L + 1, L + 2 \dots R - 1, R$ бінарів. Вам, як справжнім патріотам десяткової системи числення необхідно якнайшвидше допомогти Дециміалу, і знайти який капітал буде у нього після грошової реформи.

Технічні вимоги:

Вхідний файл: ***BINARY.IN***

Вихідний файл: ***BINARY.OUT***

Обмеження часу: 1 секунда

Формат вхідних даних: у єдиному рядку вхідного файлу ***BINARY.IN*** знаходиться два числа L та R ($1 \leq L \leq R \leq 10^{50}$). Числа відокремлені рівно одним пробілом і не мають ведучих нулів.

Формат вихідних даних: у файл ***BINARY.OUT*** запишіть одне число - суму бінарів Дециміала після грошової реформи.

Приклад вхідних та вихідних даних:

BINARY.IN	BINARY.OUT
1 10	17

Задача 6. «Ключ-2»

(Заочна олімпіада, перша ліга, 2006-2007)

Після того, як фермер Вампіров розшифрував повідомлення фермера Дістенка, звісно, він не зміг залишитися у боргу, та послав повідомлення у відповідь, яке також було зашифроване. Ключем до шифру було додатне ціле число, яке можна було отримати з надісланого N , викресливши одну цифру так, щоб число, яке залишиться було якнайбільшим. Допоможіть тепер фермеру Дістенку розшифрувати повідомлення.

Вхідні дані: у текстовому файлі **Z2.dat** записано рядок тексту довжиною не більше, ніж 100000 та не менше ніж 2, символами якого є цифри числа N .

Вихідні дані: у текстовий файл **Z2.sol** записати рядок тексту з отриманим числом.

Приклад вхідних та вихідних даних:

1)

Z2.dat	Z2.sol
321	32

2)

Z2.dat	Z2.sol
123	23

Задача 7. «Свято цукерок»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Петрика та двох його друзів пригостили цукерками. Всі цукерки розклали на три купки. Звісно, Петрик хотів би взяти собі найбільшу. Допоможіть Петрику зробити цей вибір.

Вхідні дані:

Вводяться три натуральних числа через пропуск. Кожне з чисел - кількість цукерок у купці не перевищує 10^{100} .

Вихідні дані:

Потрібно вивести одне ціле число - максимальну кількість цукерок.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
10 15 12	15

Жадібний алгоритм

Задача 1. Сума

(Заочна олімпіада, 2001-2002)

Дано n цілих чисел $x[1], x[2], \dots, x[n]$ ($1 \leq n \leq 100000$, $-10000 \leq x[i] \leq 10000$). Знайти максимальне значення суми, яку можна отримати додаванням групи послідовних чисел $x[i] + x[i+1] + x[i+2] + \dots + x[j-1] + x[j]$, ($1 \leq i \leq j \leq n$).

У вхідному потоці міститься число n , а за ним числа $x[1], x[2], \dots, x[n]$.
Всі числа розділено пропусками, потік довільно розбитий на стрічки.

У вихідний потік вивести одне число -- значення максимальної суми послідовних елементів.

Приклад.

Вхід: 10 -1 1 2 3 -1 50 -5 6 -100 50

Вихід: 56

Задачі для початківців

Задача 1. «Фотоапарат»

(Заочна олімпіада, перша ліга, 2006-2007)

При фотографуванні цифровий фотоапарат створює файл, якому автоматично надається ім'я, що відповідає шаблону F00000.jpg, де замість 00000 вказується номер знімку (наприклад, F00015.jpg). Фотограф зберігає ці файли в одній папці, а якщо редагує фотографію, то створює копію файлу, додаючи до імені символ підкреслювання "_" та цифру, яка означає номер редакції (наприклад, F00015_3.jpg - третя редакція файлу F00015.jpg).

Маючи список файлів потрібно визначити у скількох з них зберігаються відредаговані знімки.

Вхідні дані: у текстовому файлі **FOTO.DAT** записані імена файлів, відокремлені довільною кількістю пропусків.

Вихідні дані: у текстовий файл **FOTO.SOL** вивести одне ціле число - кількість файлів з відредагованими знімками.

Приклад вхідних даних - текстовий файл FOTO.DAT:

F00001.jpg F00002.jpg F00051_4.jpg F00002.jpg F00001_3.jpg
F00007.jpg F99999_8.jpg

Приклад вихідних даних - текстовий файл FOTO.SOL:

3

Задача 2. «Сніжинки в листопаді»

(Заочна олімпіада, перша ліга, 2006-2007)

В листопаді мороз формує особливі сніжинки: з кристалів, що мають форму відрізків прямої, у вигляді візерунка, що зображений на малюнку. Радіальні кристали послідовно пронумеровані від 0 до 5 і утворюють кути 60 градусів. Кристали, що утворюють вкладені шестикутники, нумеруються від центру сніжинки (сам центр має номер 0) і кріпляться до радіальних кристалів з одиничним кроком.

чоловічка, який він проходить зі свого стартового вузла до одного з платинових, обов'язково проходячи перед цим через інший платиновий вузол.

Вхідні дані: зі стандартного вхідного потоку вводяться шість чисел по два, відокремлені пропусками, в рядку – координати стартового вузла чоловічка, першого і другого платинових вузлів сніжинки відповідно.

Вихідні дані: у стандартний вихідний потік вивести одне число – довжину найкоротшого шляху кристалічного чоловічка між цими вузлами.

Приклад:

Вхідні дані:	Вихідні дані:
3 1	5
1 3	
2 2	

Задача 3. «Покер»

(Заочна олімпіада, перша ліга, 2006-2007)

Задано п'ять цілих чисел. Якщо:

- є п'ять однакових, то вивести “**Impossible**”, інакше;
- є чотири однакових, то вивести “**Four of a Kind**”, інакше;
- є однакових 3 і 2, то вивести “**Full House**”, інакше;
- є п'ять послідовних, то вивести “**Straight**”, інакше;
- є три однакових, то вивести “**Three of a Kind**”, інакше;
- є два і два однакових, то вивести “**Two Pairs**”, інакше;
- є два однакових, то вивести “**One Pair**”, інакше;
- вивести “**Nothing**”.

Обмеження: числа не перевищують 100 .

Вхідний файл: *poker.in*

В єдиному рядку записані п'ять чисел, відокремлені пропусками.

Вихідний файл: poker.out

Вивести один рядок – результат аналізу.

Приклад 1.

poker.in	poker.out
1 3 9 3 2	One Pair

Приклад 2.

poker.in	poker.out
1 5 5 4 4	Two Pairs

Приклад 3.

poker.in	poker.out
1 5 2 4 3	Straight

Задача 4. «Гноми-письменники»

(Заочна олімпіада, перша ліга, 2006-2007)

Група з 11 прогресивних гномів вирішила написати посібник з програмування для казкової школи. Щоб посібник не був занадто грубим, вони вирішили, що кожен напише не більше ніж 200 сторінок. Коли робота була закінчена, виникла потреба порахувати, скільки сторінок матиме книга. Складіть програму, щоб за відомою кількістю сторінок, які написав кожен з гномів, визначити загальну кількість сторінок.

Вхідні дані: зі стандартного вхідного потоку вводиться 11 цілих чисел - кількості сторінок, написаних кожним з учасників проекту.

Вихідні дані: у стандартний вихідний потік вивести одне ціле число - загальну кількість сторінок у книзі.

Приклад вхідних даних:

1 2 3 4 5 6 7 8 200 9 1

Приклад вихідних даних:

246

Задача 5. «Тюль»

(Заочна олімпіада, перша ліга, 2006-2007)

На фабриці, яка випускає тюль - тканину, що складається з окремих квадратних комірок, розташованих рівними рядками і стовпцями, працює відділ з розробки візерунків. Для утворення візерунку дизайнер деякі комірки зафарбовує повністю, частину комірок зафарбовує наполовину, інші залишає незафарбованими. Оптимальним вважається візерунок, у якому незафарбованих комірок стільки ж, як всіх інших разом. Потрібно скласти програму для перевірки того, чи є візерунок оптимальним.

Вхідні дані: у першому рядку текстового файлу **Z2.DAT** записані два цілі числа D та S , відокремлені пропуском - кількість комірок по довжині та ширині візерунку відповідно. У наступних D рядках записані по S чисел: 0, якщо відповідна комірка порожня, 1 - зафарбована наполовину, 2 - зафарбована повністю. Жоден з розмірів візерунку не перевищує 1000.

Вихідні дані: у текстовий файл **Z2.SOL** вивести слово, що складається з англійських літер: "yes", якщо візерунок оптимальний, "no" - в іншому випадку.

Приклад 1

Файл Z2.DAT:

```
5 6
1 1 1 1 1 1
1 0 0 0 0 1
0 0 2 2 0 0
1 0 0 0 0 1
1 1 1 1 1 1
```

Файл Z2.SOL:

no

Приклад 2

Файл Z2.DAT:

2 3

1 2 2

0 0 0

Файл Z2.SOL:

yes

Задача 6. «Кролики не тільки...»

(Заочна олімпіада, перша ліга, 2006-2007)

У фермера Вампірова система продажу кроликів базувалася на підрахунку середньої маси кролика на кролефермі. Якщо середня маса перевищувала деяке значення, яке відоме тільки фермеру, то фермер Вампіров на всіх кроликів з кролеферми дивився не як на симпатичних пухнастих тваринок, а як на 3-4 кг дієтичного м'яса. Допоможіть автоматизувати знаходження середньої маси кролика на кролефермі. Записуючи масу кролика, фермер округлював одержане число, а тому усі значення мас у нього – це натуральні числа.

Вхідні дані: у текстовому файлі **rabbit.dat** записані маси кроликів, відокремлені пропусками. Маса одного кролика не перевищує 15, а кролеферма в змозі розмістити не більше 1000 000 шт. кролів.

Вихідні дані: у текстовий файл **rabbit.sol** записати єдине ціле (округлене) число, що відповідає середній масі кролика.

Приклад:

Вхідний файл: **rabbit.dat**

1 2 3 4 6 1 5

Вихідний файл: **rabbit.sol**

3

Задача 7. «Цивілізований ринок»

(Заочна олімпіада, перша ліга, 2006-2007)

Аналізуючи ринок збуту, фермер Вампіров вирішив, що йому вже пора виходити на ринки інших районів. Добрі традиції у розведенні кроликів є у Чемеровецькому районі, а тому він задумався, якою породою кроликів завойовувати новий ринок. Різних порід на його кролефермі було всього 10. Любі друзі йому порадили: чим більша середня маса кролика у породі, тим краще. Допоможіть автоматизувати знаходження номера породи, з якою фермер буде виходити на новий ринок. Запис, який зробив фермер, має таку структуру: перше число – це маса кролика, через пропуск – номер породи, далі через пропуск записано масу наступного кролика, а через пропуск номер його породи і т.д. Фермер округлював значення мас, а тому усі значення у нього – це натуральні числа.

Вхідні дані: у текстовому файлі **market.in** через пропуск записані маси кроликів та номери порід. Маса одного кролика не перевищує 15, а кролеферма в змозі розмістити не більше 1000 000 шт. кролів.

Вихідні дані: у текстовий файл **market.out** записати одне число, що відповідає номеру породи з якою потрібно завойовувати ринок.

Приклад :

Вхідний файл: **market.in**

1 6 2 3 4 6 3 1 2 3 4 3 6 6

Вихідний файл: **market.out**

6

Задача 8. «Особливий день»

(Заочна олімпіада, перша ліга, 2006-2007)

Фермер Петро Дієтенко завдяки своєму успішному бізнесу у кролівництві став відомим далеко за межами Чемеровеччини. У Петра було дві радості у житті. Перша - це коли збільшувався приплід на власній фермі, і друга – коли відбувався падіж у конкурента Вампірова. Для контролю за

кількістю кроликів Петро кожного дня, починаючи з 1 січня, записував скільки кроликів у нього на фермі прибавилося. В кінці року він знаходив у своїх записах максимальний приплід і визначав відповідний день тижня. Цей день мав для фермера особливий статус на наступний рік і тільки кролики знали як він його проводив. Складіть програму, яка за відомими записами 2007 року визначить особливий для Петра день.

Вхідні дані: у стандартному вхідному потоці, по одному у рядку, розміщена певна кількість цілих невід’ємних чисел не більших 1000, що визначають щоденний приплід кролеферми Дієтенка. Кількість чисел не перевищує 365 (вважатимемо, що наші дії можуть відбуватися у кінці цього року).

Вихідні дані: у стандартний вихідний потік вивести скорочену англійську назву особливого дня. Наприклад: Mon, Tue, Wed, Thu, Fri, Sat, Sun (Перелік днів іде від понеділка до неділі в порядку їх слідування) . Якщо днів із максимальним приплодом є декілька, то вивести той, що є першим у переліку.

Приклад вхідних даних.

2

9

3

Приклад вихідних даних

Tue

Примітка. Найбільше значення у переліку 9. Його порядковий номер 2 і це є другий день 2007 року, що припадає на вівторок.

Задача 9. «Міні-шахи»

(Заочна олімпіада, перша ліга, 2006-2007)

Оскільки фермери Дієтенко та Вампіров постійно опікуються своїми кроликами, то вільного часу у них вистачає лише на гру в міні-шахи. Дошка для гри в міні-шахи стандартна: стовпці нумеруються великими латинськими

літерами від 'А' до 'Н', рядки - цифрами від 1 до 8; нумерація розпочинається з нижнього лівого кута. Відмінності від звичайних шахів:

а) фігур значно менше (король і два пішаки одного кольору та слон іншого);

б) пішаки можуть стояти і в крайніх рядках.

Допоможіть постійно зайнятим фермерам швидко визначити, скільком фігурам загрожує слон.

Вхідні дані: у першому рядку текстового файлу **chess.in** записані два



символи - літера рядка і номер стовпця, на перетині яких знаходиться слон, у наступних трьох рядках, по 2 символи у кожному - координати короля та двох пішаків відповідно.

Вихідні дані: у перший рядок текстового файлу **chess.out** вивести одне число - кількість фігур, яким загрожує слон, або **NO**, якщо слон не загрожує жодній фігурі.

Приклад файлу chess.in:

C3

A1

D2

D3

Приклад файлу chess.out:

2

Задача 10. «Підкидний»

(Заочна олімпіада, перша ліга, 2006-2007)

В міні-шахах сталась патова ситуація, і фермери вирішили пограти в карти.

Карта має дві характеристики: ранг R (6,7,8,9,10,J,Q,K,A) і масть M (P, T, C, B). Карти мають старшинство відповідно рангу: шістка (6) - наймолодша карта, туз (A) - найстарша. Покрити карту можна або старшою картою такої ж масті, або картою козирної масті, якщо карта, яку криють, сама не є козирною - в такому випадку її можна покрити лише старшим козирем..

Визначити, чи криє карта (R1,M1) карту (R2,M2), якщо козир K.

Вхідні дані: у першому рядку файлу **CARDS.DAT** записаний символ K - козирна масть; у наступних рядках - символи R1, M1, R2, M2, кожен в окремому рядку - ранг і масть першої та другої карт відповідно.

Вихідні дані: у перший рядок текстового файлу **CARDS.RES** вивести **YES**, якщо перша карта криє другу, або **NO** в протилежному випадку.

Приклад файлу CARDS.DAT:

P

A

T

6

C

Приклад файлу CARDS.RES:

NO

Комбінаторика

Задача 1.

(IV обласна олімпіада, 1990 рік)

Жителі деякого племені говорять на якомусь країн ном мовою: будь-яке промовлене слово вони утворюють з усіх букв деякого алфавіту, що складається з n букв. Слова, в яких порядок проходження літер збігається, вважаються однаковими, якщо ж у двох словах порядок проходження букв не збігається, то такі слова вважаються різними.

Старійшини цього племені доручили самому мудрому "філологу" скласти словник усіх слів мови, на якому вони говорять. "Філолог" взявся за роботу. Не минуло й 20 років, як такий словник був складений. Старійшини залишилися задоволені, але сам "філолог" не був упевнений, що всі можливі слова включені до його словника.

Чи можете Ви впоратися з таким самим завданням, але так, щоб напевне до словника племені були занесені всі різні слова, що вживаються жителями?

Задача 2. «Перестановка»

(XVII обласна олімпіада, 2003 рік)

Таблицею інверсій заданої перестановки чисел від 1 до N є масив, K -й елемент якого рівний кількості елементів більших за K , що стоять у перестановці зліва від K . Так для перестановки 3 1 5 2 4 таблицею інверсій буде: 1 2 0 1 0. Потрібно побудувати алгоритм і реалізувати його у вигляді програми PERM.PAS (BAS чи CPP), що прочитає вхідні дані з файлу PERM.DAT, за даною таблицею інверсій відновить перестановку та виведе її у файл PERM.SOL

У першому рядку файлу PERM.DAT записано число N .

У другому - N чисел (таблиця інверсій), розділених пропуском.
($1 \leq N \leq 100$)

У файл PERM.SOL необхідно записати шукану перестановку.

Приклад:

PERM.DAT

5

1 2 0 1 0

PERM.SOL

3 1 5 2 4

Задача 3. «Олімпіада і хаос»

(XX обласна олімпіада, 2006 рік)

Для проведення змагань з програмування виділено N робочих місць, які пронумеровані числами від 1 до N . Кожному учаснику змагань виданий номер його робочого місця. Після того, як всі учасники зайняли робочі місця виявилось, що ніхто з учасників не сидить на своєму місці. Журі зацікавило питання: скільки існує варіантів розміщення учасників на робочих місцях таких, що ніхто не знаходиться на своєму місці. Допоможіть журі знайти відповідь на це питання.

Вхідні дані: в першій стрічці вхідного потоку міститься одне число – кількість робочих місць. Це число більше від 0 та не перевищує 100.

Вихідні дані: в першу стрічку вихідного потоку вивести одне число – кількість можливих розміщень учасників на робочих місцях, таких, що ніхто з них не знаходиться на своєму місці.

Приклад вхідних даних:

3

Приклад вихідних даних, що відповідають вхідним:

2

Задача 4.

(XXIII обласна олімпіада, 2009 рік)

Виведіть всі правильні вирази, що складаються з круглих та квадратних дужок довжиною N.

Задача 5.

(Заочна олімпіада, 2000-2001)

Дано поле в клітинку розміром $3 \times N$. Підрахуйте, скількома способами можна цілком заповнити дане поле прямокутниками розміром 2×1 .

Вхідні дані: в текстовому файлі 'POLE.DAT' в першому рядку число N ($0 \leq N \leq 60$).

Вихідні дані: в перший рядок текстового файлу 'POLE.SOL' вивести кількість комбінацій розташування прямокутників на даному полі.

Приклад вхідних даних №1:

2

Приклад вихідних даних №1:

3

Приклад вхідних даних №2:

3

Приклад вихідних даних №2:

0

Задача 6. «Номер перестановки»

(Заочна олімпіада, 2000-2001)

Візьмемо всі можливі рядки, які можна побудувати з символів "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", якщо брати кожен з них по одному разу. Впорядкуємо ці рядки за алфавітом і пронумеруємо їх по порядку починаючи з одиниці. Першою буде йти рядок 0123456789AB, за нею 0123456789BA, потім 012345678A9B, і т.д. , Останньої ВА9876543210
Написати програму, яка видає номер рядка в цьому списку. Вхідні дані: у

першому рядку вхідного потоку міститься 12-ти символний рядок. Результати: вивести у вихідний потік ціле число, рівне номеру цього рядка.

Задача 7.

(Заочна олімпіада, 2000-2001)

Припустимо, що ім'я файлу може складатися тільки з маленьких літер латинського алфавіту, цифр 0 .. 9 і знака підкреслення "_", а також мати довжину від 1 до 8 символів. Маска файлу може в добуток мати символи "?" і "*" і її довжина теж не повинна перевищувати 8 символів (Якщо довжина маски буде більше 8 символів, то можуть початися якісь аномальні явище) "?" означає, що на даному місці повинен бути один з допустимих символів. "*" Означає, що на даному місці може знаходитися будь-яка допустима послідовність символів. За заданою масці визначте, скільки імен файлів може їй відповідати.

Вихідні дані: у першому рядку вхідного потоку міститься маска.

У перший рядок вихідного потоку необхідно вивести шукану кількість імен файлів.

Задача 8. «Наступна перестановка»

(Заочна олімпіада, вища ліга, 2005-2006)

Дана стрічка, що складається з великих латинських літер. Довжина стрічки від 2 до 255 символів. Стрічка містить щонайменше два різних символи. Уявімо собі всі стрічки, які можна утворити шляхом переставляння символів в даній стрічці. Уявімо собі список, що містить всі такі стрічки за алфавітним порядком, та не містить двох однакових стрічок. Звичайно дана стрічка теж міститься в цьому списку.

Завдання: напишіть програму, що визначає, яка стрічка буде наступною за даною у цьому алфавітному списку. Якщо дана стрічка є останньою у цьому списку, то слід вивести першу стрічку списку.

Формат вхідних даних: вхідний потік містить єдину дану стрічку.

Формат вихідних даних: вивести в вихідний потік єдину стрічку, яку повинна знайти програма.

Приклад вхідних даних:

FDHF

Приклад вихідних даних, що відповідають вхідним:

FFDH

Задача 9. «Інверсія»

(Заочна олімпіада, перша ліга, 2006-2007)

Школяр Вася Пупкін виписав на аркуші паперу всі перестановки перших N натуральних чисел і вирішив підрахувати, в скількох виписаних перестановках число інверсій дорівнює K . Напишіть програму, яка допоможе йому це зробити.

Примітка. В перестановці $a[1], a[2], \dots, a[n]$ перших n натуральних чисел $a[k]$ і $a[m]$ утворюють інверсію, якщо $a[k] > a[m]$ і $k < m$.

Вхідні дані. в єдиному рядку файлу **inverse.dat** знаходяться числа N, K ($1 \leq N \leq 18, 0 \leq K \leq 1000$).

Вихідні дані. в файл **inverse.sol** потрібно вивести кількість перестановок, в яких рівно K інверсій.

Приклад вхідних і вихідних даних:

inverse.dat

4 2

inverse.sol

5

Задача 10. «Бірка»

(Заочна олімпіада, перша ліга, 2006-2007)

У короля Лапландії є в наявності різнокольорові стрічки (всього M різних кольорів). Він замислився (і таке буває) чи вистачить йому цих стрічок, щоб однозначно позначити воїнів королівської гвардії. Допоможіть

королю задовольнити цю забаганку. Бірка, яку отримує воїн, містить по одній смужці всіх кольорів, а порядок слідування кольорів у кожного з воїнів – унікальний.

Обмеження: $2 \leq M \leq 8$; кольори позначені символами латинського алфавіту.

Вхідний файл: *birka.in*

У єдиному рядку якого, записаний рядок, кожний з M символів якого позначає колір однієї стрічки.

Вихідний файл : *birka.out*

У перший рядок записати, число яке визначає, яку кількість воїнів можна однозначно позначити, а далі у кожний рядок вихідного файлу записати по одній позначці.

Приклад:

birka.in	birka.out
IOX	6
	XOI
	OIX
	IXO
	XIO
	OXI
	IOX

Задача 11. «Спіраль»

(Заочна олімпіада, вища ліга, 2006-2007)

Спіраль – це ламана не нульової довжини без самоперетинів, вершини якої розташовані у точках з цілими координатами на площині. Кожна наступна ланка ламаної повинна бути повернута відносно попередньої на 90° за годинниковою стрілкою.

Розглянемо прямокутник N на M ($1 \leq N, M \leq 20$). Нехай для кожної спіралі її перша вершина співпадає з лівою верхньою вершиною прямокутника, а друга лежить на верхній стороні прямокутника.

Новий Прем'єр-міністр в умовах енергетичної кризи хоче налагодити випуск спіралей. Його цікавить кількість таких спіралей.

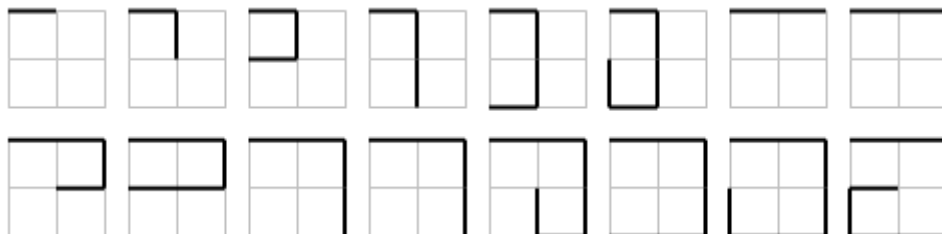
Напишіть програму, яка визначає кількість таких спіралей, що лежать в межах даного прямокутника.

Формат вхідних даних: текстовий файл **SPIRAL.DAT** містить два цілих числа N та M .

Формат вихідних даних: текстовий файл **SPIRAL.SOL** має містити одне число – кількість знайдених спіралей.

Приклад вхідних та вихідних даних:

SPIRAL.DAT	SPIRAL.SOL
2 2	16



Задача 12. «Рядок»

(Заочна олімпіада, 2008-2009)

Ім'я вхідного файлу:

string.in

Ім'я вихідного файлу:

string.out

Дано слово, Вам необхідно в ньому переставити місцями два символи, які знаходяться на різних позиціях в цьому слові, так щоб в результаті отримати слово, лексикографічно найменше з усіх можливих.

Формат вхідних даних: у єдиному рядку вхідного файлу записане слово, яке складається з маленьких літер латинського алфавіту. Гарантується, що його довжина не буде перевищувати 100000 і не буде меншою за 2 символи.

Формат вихідних даних: у єдиному рядку вихідного файлу повинно знаходитись рівно два числа – позиції символів у даному слові, що треба переставити між собою. Якщо існує декілька варіантів відповіді – виведіть ту в якій менше перше число, якщо таких більше однієї, то ту в якій менше друге число.

Приклад файлів вхідних і вихідних даних:

string.in

adbsab

string.out

2 5

Задача 13. «Доміно своїми руками»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 0.5 с

Ліміт пам'яті: 64 Мбт

На олімпіаду з трудового навчання Василько вирішив виготовити кості для гри в доміно (шахи Василько виготовляв уже минулого року). Креативність учня не дозволяла йому взятися за виготовлення стандартного набору із максимальною кількістю точок на одній половині кості рівною 6. Виготовлення точок на костях виявилось достатньо трудомісткою роботою і Василько вирішив підрахувати наперед скільки точок будуть мати нестандартні набори доміно. Наприклад, якщо максимальна кількість точок буде рівна 2, то треба буде зробити всього 12 точок на комплект. Малувато для задуманого проекту... Якщо взяти 3, то... Василько втомився рахувати і написав програму, яка визначає кількість точок у різних наборах доміно. Попробуйте і ви написати програму, яка буде визначати кількість точок для

різних наборів костей доміно із максимальною кількістю точок не більшою N ($1 \leq N \leq 10000$).

Вхідні дані:

Стандартний вхідний потік містить число N .

Вихідні дані:

У стандартний вихідний потік вивести кількість точок відповідного набору доміно.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
2	12

Пояснення:

Даний набір буде містити такі кості: 0-0, 0-1, 0-2, 1-1, 1-2, 2-2. Всього є 12 точок.

Масиви

Задача 1.

(I обласна олімпіада, 1987 р.)

В прямокутній цілочисельній таблиці $A[1:100,1:50]$ всі числа різні. В кожному рядку вибирається максимальний елемент, потім серед цих чисел вибирається мінімальний. За допомогою алгоритму, записаного на алгоритмічній мові, знайти це мінімальне число та номер рядка в якому воно знаходиться.

Задача 2.

(II обласна олімпіада, 1988 рік)

Фрагменти таблиці $X[1:N]$ являють собою послідовність її елементів $X[k], X[k+1], X[k+2], \dots, X[l-1], X[l]$, розміщених послідовно в таблиці X , при $1 \leq k \leq l \leq N$. Написати алгоритм, який визначає початковий та кінцевий номери самого довгого монотонного фрагмента таблиці X . Монотонним фрагментом назовемо такий фрагмент, в якому значення X монотонно зростають з ростом індекса або монотонно спадає або рівні між собою.

Задача 3.

(II обласна олімпіада, 1988 рік)

Дано таблиці $X [1:100]$ і $Y [1:100]$. Записати алгоритм, що міняє послідовно місцями значення елементів $X [k]$ і $Y [k]$ цих таблиць для $k=1, \dots, 100$ за наступним правилом: $X [k]$ приймає значення меншого з $X [k]$, $Y [k]$, а $Y [k]$ приймає значення більшого з цих елементів.

Задача 4.

(II обласна олімпіада, 1988 рік)

Дана цілочисельна таблиця A . Необхідно переставити на початок таблиці всі негативні елементи, не порушуючи порядок їх слідування у вихідній таблиці.

Задача 5.

(IV обласна олімпіада, 1990 рік)

Дана таблиця n цілих чисел. Розробити алгоритм, що дозволяє знайти в цій таблиці "симетричну" частину найбільшої довжини послідовності елементів.

У "симетричній" частині перший елемент дорівнює останньому, другий – передостанньому і т.д.

Задача 6. «Заповнення по діагоналях»

(V обласна олімпіада, 1991 рік)

Скласти алгоритм заповнення двомірної таблиці $A[1:n, 1:n]$ по діагоналях з північного сходу на південний захід числами $1, 2, \dots, n^2$, починаючи з північно-західного кута таблиці.

Задача 7. «Циклічна перестановка»

(VI обласна олімпіада, 1992 рік)

Скласти алгоритм циклічної перестановки елементів таблиці $B[1:100]$, при якій $B[i]$ переміщується в $B[i+1]$, а $B[100]$ переміщується в $B[1]$.

Задача 8. «Різні числа»

(VII обласна олімпіада, 1993 рік)

Дано числову таблицю $A[1:m]$. Підрахувати, скільки різних чисел в цій таблиці. Наприклад, в таблиці $5, 7, 5, 4$ різних чисел три ($5, 7$ і 4).

Задача 9. «Симетричний фрагмент таблиці»

(VII обласна олімпіада, 1993 рік)

Задано таблицю натуральних чисел $A[1:N]$. Знайти в цій таблиці фрагмент максимальної довжини, в якому перше число дорівнює останньому, друге – передостанньому і т.д. Надрукувати цей фрагмент таблиці.

Задача 10. «Особливі елементи»

(VII обласна олімпіада, 1993 рік)

Задано прямокутну таблицю $B[1:m, 1:n]$. Один з елементів цієї таблиці назвемо "особливим", якщо він є одночасно найбільшим у своєму рядку і найменшим у своєму стовпчику. Знайти всі "особливі" елементи таблиці, або вивести "ні", якщо їх немає.

Задача 11. Обробка фотографій з космосу

(VIII обласна олімпіада, 1994 рік)

Комп'ютер перетворив фотографію частини поверхні планети, зроблену з космічного апарата, в двомірну таблицю $Map[1:M, 1:N]$, елементи якої - цифри 0 (суша) та 1 (вода).

Є 4 типи поверхні суші (материк, острів, півострів та берег) та три типи поверхні води (море, озеро та затока). Тип кожної клітини можна визначити в залежності від того, до яких типів було однозначно віднесено раніше деякі з чотирьох сусідніх по вертикалі та горизонталі клітин:

Тип 2 - материк-клітина суші, оточена чотирма клітинами суші.

Тип 3 - острів-клітина суші, оточена чотирма клітинами води.

Тип 4 - півострів-клітина суші, що має 3 сусідніх клітини води, або 2 сусідніх клітини води та ≥ 1 клітини півострова, або 1 сусідню клітину води та ≥ 2 клітин півострова.

Тип 5 - берег-клітина суші, що не належить до типів 2-4.

Тип 6 - море-клітина води, що має хоча б одну сусідню клітину моря.

Тип 7 - затока-клітина моря, що має 2 або 3 сусідніх клітини суші, або 4 сусідні клітини затоки, або 1 сусідню клітину суші та не менше 2 клітин затоки.

Тип 8 - озеро-клітина води, що не належить до типів 6, 7.

Вважається, що дві клітини типу 4 належать до одного півострова, якщо між ними можна пройти по суші, переходячи на сусідні клітини типу 4 по горизонталі або вертикалі. Таким же чином дві клітини типу 7 належать

до однієї затоки, якщо між ними можна проплисти по воді, перепливаючи на сусідні клітини типу 7 по горизонталі або вертикалі. Дві клітини типу 8 належать до одного озера, якщо між ними можна проплисти по воді, перепливаючи на сусідню клітину типу 8 по горизонталі або вертикалі. Крайні рядки та стовпчики таблиці займає море.

ЗАВДАННЯ: написати програму, що для заданої таблиці-карти:

1. Визначить номер типу кожної її клітини.
2. Обчислить кількості островів, півостровів, озер та заток на карті.

Задача 12.

(X обласна олімпіада, 1996 рік)

Курси K іноземних валют по відношенню до українського карбованця за N днів містяться в таблиці S . $S[i, j]$ - курс валюти j в день i . На скільки відсотків максимально можна було б збільшити за ці дні капітал, граючи на зміні курсів цих валют? Гроші можна як завгодно розподіляти в карбованці та іноземну валюту. Початковий капітал - в карбованцях; кінцевий також повинен бути в карбованцях. Курси продажу і покупки вважаються однаковими. Довгу арифметику можна не реалізовувати. Суми грошей вважайте дійсними числами.

Якщо задача розв'язується практично, і комп'ютер дозволяє працювати з файлами, передбачте такий формат даних. Вхідні дані знаходяться в файлі CURRENCY.INP. Перший рядок файлу містить кількість тестів. Перший рядок кожного тесту містить 2 числа: кількість валют K та кількість днів N . Наступні N рядків по K натуральних чисел - курси валют. Для кожного тесту обчислити вказаний відсоток та вивести його у файл CURRENCY.OUT. При неможливості роботи з файлами забезпечте ввід даних з клавіатури.

Задача 13. «Мікроорганізми»

(XI обласна олімпіада, 1997 рік)

Для обробки фотознімків мікроорганізмів, виконаних під мікроскопом, кожну фотографію розділено на дрібні клітинки. В кожній клітинці, яка повністю накрита одним з мікроорганізмів, або в якій міститься частина мікроорганізму, зроблено позначку.

Вважається, що дві клітинки з позначками належать одному й тому ж мікроорганізму, якщо з однієї з них можна потрапити в іншу, рухаючись по клітинках з позначкою ліворуч, праворуч, вгору або вниз.

Дано: прямокутне фото розміром $m \times n$ клітинок, частину яких позначено.

Отримати: кількість організмів на фотознімку.

Задача 14. «Футбольна таблиця»

(XVI обласна олімпіада, 2002 рік)

У першому рядку файлу FOOTBALL.DAT містяться два числа, N - кількість команд та M - кількість зіграних матчів. У наступних N рядках - назви команд. Кожен матч описаний у файлі *.DAT, де * - порядковий номер матчу. (1.DAT, 2.DAT, ...). Файл що описує кожен матч має таку структуру: Склад команд описується двома послідовними блоками. У першому рядку блоку записана назва команди, а у другому - число P - кількість гравців, що були заявлені на матч цією командою. У наступних P рядках - прізвища гравців. Далі записано число K - кількість забитих у матчі голів. Наступні K рядків містять прізвища гравців, що забили голи.

Побудувати турнірну таблицю за правилами: Команда, що набрала більше очок, повинна йти раніше у таблиці. При рівності очок, першою буде йти команда яка має більше перемог, а якщо цей параметр теж буде рівний, то впорядкування повинно бути за алфавітним порядком. Записати у файл TABLE.REP кожний рядок таблиці таким чином: Спочатку друкується назва команди, далі йде N блоків такого формату "A:B", " - " або " X ". Нехай номер

поточної команди Num1, а номер поточного блоку Num2. " X " - ставиться, якщо Num1 = Num2. "A:B" - результат гри між командами Num1 та Num2. (A, B - кількість голів забитих командою Num1 та Num2 відповідно). " - " - якщо Num1 та Num2 ще не грали між собою. В кінці рядка - загальна кількість очок, які набрала команда. Перший блок повинен починатися в рядку з позиції під номером 25. За перемогу нараховується 3 очка, за нічию 1, за програш 0. Обмеження: $2 \leq N, P \leq 20$; $0 \leq K \leq 10$. Назви команд та прізвища гравців складаються не більше 20 латинських літер та цифр. Команди грають між собою не більше одного матчу.

FOOTBALL.DAT

3 2

BALLERS

FOOTERS

PROGRAMMERS

1.DAT

FOOTERS

4

Player1

Player2

Player3

Player4

BALLERS

5

Gamer1

Gamer2

Gamer3

Gamer4

Gamer5

3

Player1
GAMER3
PLAYER3

2.DAT
PROGRAMMERS

4

Olymp1

Olymp2

Olymp3

Olymp4

FOOTERS

3

Player1

Cook

Lamberto

2

Olymp3

Olymp2

TABLE.REP

FOOTERS X 0:2 2:1 3

PROGRAMMERS 2:0 X - 3

BALLERS 1:2 - X 0

Задача 15.

(Заочна олімпіада, 2000-2001)

Дано розмірність квадратної матриці N , яку заповнили числами від 1 до N^2 по спіралі.

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

З'ясуйте, яке число знаходиться в позиції [R, C], де R - номер рядка, C - номер стовпчика.

Вхідні дані: перший рядок - число N ($1 < N \leq 100000$); в другому рядку через пропуск числа R і C.

Вихідні дані: відповідне число.

Приклад вхідних даних:

6

2 1

Приклад вихідних даних:

20

Задача 16. «Зигзаг»

(Заочна олімпіада, 2001-2002)

Квадратну матрицю розмірністю $N \times N$ ($N \leq 40000$) заповнено натуральними числами наступним чином:

для $N=4$

1 3 4 10

2 5 9 11

6 8 12 15

7 13 14 16

для $N=5$

1 3 4 10 11

2 5 9 12 19

6 8 13 18 20

7 14 17 21 24

15 16 22 23 25

За даними натуральними числами $L \leq N$ (номер рядка) і $C \leq N$ (номер стовпчика) вказати число, яке знаходиться на перетині L-го рядка і C-го стовпчика.

Вхідні дані: в файлі **INPUT.DAT** знаходиться три числа N, L і C (через пропуск).

Вихідні дані: вивести у файл **OUTPUT.DAT** число, яке знаходиться на перетині L-го рядка і C-го стовпчика.

<i>Приклад 1.</i> INPUT.DAT 4 3 3	<i>Приклад 2.</i> INPUT.DAT 40 40 40	<i>Приклад 3.</i> INPUT.DAT 43210 1234 5678
---	--	---

OUTPUT.DAT 12	OUTPUT.DAT 1600	OUTPUT.DAT 23878739
------------------	--------------------	------------------------

Задача 17.

(Заочна олімпіада, 2001-2002)

Математику Дифуру Інтеграловичу необхідно перемножити N прямокутних матриць.

Відомо, що для обчислення добутку двох матриць розмірами $M \times N$ та $N \times K$ необхідно виконати $M * N * K$ елементарних операцій, а результуюча матриця має розмір $M \times K$.

Матриці-множники не можна міняти місцями, але можна змінювати порядок множення матриць, наприклад:

при обчисленні $A(5 \times 50) * B(50 \times 2) * C(2 \times 1)$

забороняється $A * C * B$, але

не забороняється $A * (B * C)$

У випадку $A * B * C$ кількість операцій $500 + 10 = 510$, а

у випадку $A * (B * C)$ кількість операцій $100 + 250 = 350$.

Напишіть програму, яка для заданих матриць вкаже, в якому порядку необхідно виконати операції множення.

Вхідні дані:

В першому рядку файлу MATRIX.DAT записане число N – кількість прямокутних матриць.

В наступних N рядках по два числа через пропуск: розміри матриць.

Кількість матриць і їх розміри не перевищують числа 50.

Вихідні дані:

В $N-1$ рядках файлу MATRIX.SOL по одному числу – номера (початкові) операторів множення.

Задача 18. «Майданчик під стадіон»

(Заочна олімпіада, 2002-2003)

Жителі невеликого провінційного містечка Healthcity, що розміщено не вельми далеко від кінця світу любили дві речі: спорт і природу. Вирішивши одного разу побудувати стадіон вони були шоковані - навколо містечка знаходилося безліч дерев, які вони, по зрозумілих причинах, не могли нищити. Пошук придатного для будівництва майданчика затягнувся і мер звернувся до аерокосмічного агентства за планом місцевості з відмітками місць, де ростуть дерева. План було отримано. Він був розбитий на квадрати розмірності N , які могли мати такий вигляд:

0 0 -1 -1 0 -1

0 -1 -1 -1 1 2

-1 -1 0 -1 3 4

-1 0 4 4 -1 -1

-1 -1 6 4 -1 -1

0 0 -1 0 0 -1

Від'ємними одиничками позначені місця знаходження дерев, а додатні числа вказують на вільні від дерев місця та середню висоту ділянки відносно найнижчої точки на карті. Ваша задача: допомогти жителям міста Healthcity поліпшити здоров'я і зберегти природу, тобто скласти програму вибору найбільшого майданчика для майбутнього стадіону. Ділянка повинна мати прямокутну форму (квадрат є частковим випадком прямокутника). З двох рівних по площі ділянок слід вибрати ту, затрати праці по підготовці якої будуть мінімальними.

Наприклад, для нашого плану рівні площі мають дві ділянки: $(2,5)$ і $(4,3)$. Координати задають верхні ліві кути. Площі однакові (4 од), але другий варіант кращий, оскільки зайві тільки 2 одиниці ґрунту (координата $5,3$), а в першому випадку для виходу на цей рівень ми повинні перемістити 1 од. з $(3,5)$ в $(2,5)$. І хоча тепер перша рівнина $(2,2,2,4)$ дорівнює по об'єму

робіт другій (4,4,6,4) - друга більш економічно вигідна тому, що тут не здійснювалися попередні переміщення.

Приклади порівняння рівнин:

(2,2,2,3) краще, ніж (2,2,2,1) - одиницю ґрунту краще вивести як завезти;

(1,2,3,4) дорівнює (5,6,7,8) - в такому випадку виводити першу знайдену при обході з верхнього лівого кута карти.

Технічні умови:

Вхідний текстовий файл STADION.DAT у першому рядку містить число $0 < N < 101$. У наступних N рядках міститься по N цілих чисел через пропуск.

Вихідний файл STADION.SOL у першому рядку містить два цілих числа розділених пропуском - координати лівого верхнього кута прямокутньої ділянки; другий рядок містить одне ціле число - площу ділянки; третій одне ціле число - кількість одиниць ґрунту для переміщення.

Для нашого прикладу вихідний файл повинен містити:

4 3

4

2

Задача 19. «Satellite Photographs»

(Заочна олімпіада, вища ліга, 2005-2006)

Новий український фермер, обтяжений грішми та надлишковою освітою, бажаючи отримувати надвисокі прибутки замовив космічному агентству супутникові фотографії своєї ферми та достатньо обширних земель, на яких він збирався випасати худобу. Його цікавило найбільш неперервне пасовище. Пасовище назвемо неперервним, якщо його точки можуть бути з'єднанні вертикально або горизонтально. Кожне фото розміром $W \times H$ пікселів (є таке слово в українській мові) складається із символів "*" – пасовище та "." не пасовище.

Вхідні дані:

В першому рядку вхідного файлу **Satpix.in** міститься два натуральних числа W і H ($1 \leq W \leq 80$, $1 \leq H \leq 1000$), розділених пропуском. Наступні H рядків містять опис фото: кожен рядок складається із W символів "*" або ".".

Вихідні дані:

Файл **Satpix.out** містить єдине число – величину найбільшого неперервного пасовища.

Наприклад:

Satpix . in	Satpix . out
10 5 *** **** *** *** ***	16

Задача 20.

(Заочна олімпіада, перша ліга, 2005-2006)

Рота чисельністю в N солдат вишикувалась на плацу в стрій по M солдат в шерензі. Стрій складається з шеренг та колон. Остання шеренга може бути не повною. В кожній колоні визначили найвищого солдата, а серед них – найнижчого. Його кодове ім'я – **S-007**. В кожній шерензі визначили найнижчого, а серед них – найвищого солдата. Його кодове ім'я – **S-001**. Визначити, хто вищий на зріст – найнижчий серед високих (**S-007**) чи найвищий серед низьких (**S-001**).

Вхідні дані: У першому рядку текстового файлу **rota.dat** записані два цілі числа, відокремлені пропуском: N – кількість солдат у роті, та M – кількість солдат у шерензі. В наступних рядках записані по M цілих чисел, розділених пропусками – зріст солдат чергової шеренги в сантиметрах.

Вихідні дані: У текстовому файлі **rota.sol** записане кодове ім'я солдата, вищого на зріст серед відібраних солдат, або число 0, якщо його неможливо визначити.

Приклад 1:

rota.dat	rota.sol
12 4	S-007
168 175 181 172	
193 165 170 184	
183 177 174 168	

Приклад 2:

rota.dat	rota.sol
11 3	0
190 190 190	
190 190 190	
190 190 190	
190 190	

Задача 21. «Грядки»

(Заочна олімпіада, перша ліга, 2006-2007)

Садова ділянка, що має прямокутну форму та розділена на квадратні клітинки зі стороною 1 метр, має ширину **n** і довжину **m** метрів. На цій ділянці скопані грядки прямокутної форми. Ніякі дві грядки не перетинаються й не торкаються одна одної ні вертикальною, ні горизонтальною сторонами клітинок (торкання грядок кутами клітинок допускається).

Підрахуйте кількість грядок на садовій ділянці.

Задані обмеження:

Розміри ділянки задовольняють умовам: $1 \leq n, m \leq 100$.

Вхідні й вихідні дані:

У першому рядку вихідного файлу **Z2.dat** задані два числа: **n** та **m**, розділені одним або декількома пропусками.

Далі слідує **n** рядків по **m** символів. Символ «1» позначає територію грядки. Символ «0» відповідає нескопаній території. Інших значень у вхідному файлі немає.

Вивести у вихідний файл **Z2.sol** кількість грядок на садовій ділянці.

Приклад вхідних і вихідних даних:

Z2.dat

5 6

010000

001110

001110

000000

111100

Z2.sol

3

Задача 22. «День Святого Миколая»

(Заочна олімпіада, перша ліга, 2006-2007)

До дня Святого Миколая залишилось трохи більше місяця. Відроджуючи традиції, і задля реклами свого свята, Святий Миколай влаштував змагання з Дідом Морозом, хто заробить більше дитячих усмішок. Звичайно, усмішки діти віддають взамін подарунків. Одного ранку дітки мають гарний настрій і дарують більше усмішок, а іншого ранку їх змушують застеляти ліжко – і усмішок, відповідно, менше, але кожного ранку кількість усмішок за один подарунок не менше 1 і не більше 100. Стартує Святий Миколай вранці з одного подарунка, але кожної ночі він встигає виготовити ще 1 (подарунки можна роздавати лише вранці, а виготовляти лише вночі). Змагання довгострокове і термін його закінчення Святий Миколай залишив в

таємниці, повідомивши його (термін) лише своїм друзям, однак зауважив, що змагання не триватимуть довше 2000000 діб. Друзі підготували йому прогноз кількості дитячих усмішок за один подарунок на кожен ранок відповідного періоду. Миколай не зобов'язаний дарувати подарунки щоранку. Допоможіть Святому Миколаю роздати подарунки так, щоб отримати за цей період якнайбільше усмішок.

Вхідні дані: У текстовому файлі **Usmishka.dat** знаходиться послідовність цілих чисел, записаних через пропуск – прогноз кількості усмішок, які можна отримати за один подарунок, кожного ранку періоду змагань.

Вихідні дані: У текстовому файл **Usmishka.sol** вивести одне ціле число – кількість усмішок, які отримав Святий Миколай за весь період змагання.

Приклад 1:

Usmishka.dat	Usmishka.sol
3 5 2	12

Приклад 2:

Usmishka.dat	Usmishka.sol
2 1 1 1 1	6

Задача 23. «Таблиця»

(Заочна олімпіада, перша ліга, 2006-2007)

Елементи квадратної таблиці нумеруються таким чином:

	j				
i	1	2	5	10	...
	4	3	6	11	...
	9	8	7	12	...

16	15	14	13	...
...

За заданим k ($1 \leq k \leq 211600000$) знайти його місце в таблиці i, j .
 Нумерація рядків і стовпців таблиці починається з одиниці.

Вхідні дані: зі стандартного вхідного потоку вводиться задане число k .

Вихідні дані: у стандартний вихідний потік вивести два числа – номери рядка та стовпця таблиці, на перетині яких знаходиться число k .

Приклад:

Вхідні дані:	Вихідні дані:
12	3 4

Задача 24. «Проблема королів»

(Заочна олімпіада, вища ліга, 2006-2007)

На шаховій дошці розміром N на M ($3 \leq N, M \leq 30$) стоять два одиноких королі. Наші королі незвичайні шахові, а загадкові і тому рухаються одночасно за заданою для кожного програмою. Але початкове положення кожного короля не є фіксованим і вони не співпадають.

Напишіть програму, яка для кожної клітинки поля визначає кількість розміщень другого короля, для випадку коли вони б'ють один одного, якщо перший знаходиться в даній клітинці, жоден з королів не може залишати межі поля.

Будемо вважати, що королі б'ють один одного, якщо в деякий момент часу вони потрапили на одну клітинку. Після того, як королі побили один одного, їхній рух призупиняється.

Формат вхідних даних: текстовий файл **KINGS.IN** містить три рядки: у першому записані два цілих числа N та M – розміри шахової дошки; у другому програма - послідовність ходів першого короля; у третьому програма для другого короля.

Оскільки королі загадкові, то програма містить не більше 5 команд, кожна з яких це символ, що вказує переміщення на одну з 8 сусідніх клітинок:

- U – на одну клітинку вгору;
- E – на одну клітинку вверх-вправо;
- R – на одну клітинку вправо;
- F – на одну клітинку вниз-вправо;
- D – на одну клітинку вниз;
- G – на одну клітинку вниз-вліво;
- L – на одну клітинку вліво;
- H – на одну клітинку вліво-вверх.

Формат вихідних даних: текстовий файл **KINGS.OUT** має містити **N** рядків по **M** чисел у кожному. Кожне число – це відповідь на поставлену задачу для даної клітинки. Якщо для даної клітинки послідовність ходів першого короля завжди призводить до виходу за межі дошки, то вивести -1.

Приклад введення і виведення:

KINGS.IN	KINGS.OUT
3 3	1 1 -1
DDR	1 1 -1
GDD	-1 -1 -1

Задача 25. «Кролики-3»

(Заочна олімпіада, перша ліга, 2006-2007)

Один український фермер, не втрачаючи надії розбагатіти, розводив кроликів. Кролики утримуються в огорожі, огорожа поділена горизонтальними перегородками на клітки. В кожній окремій клітці кролик вільно може пересуватися і, за наявності нори, переходити в іншу клітку. Нумерація кліток починається зверху, кожен рядок є однією кліткою і в кожній клітці може бути не більше однієї нори. У кролика є бажання поїсти

моркви, яка знаходиться завжди у правому нижньому, на малюнку, кутку огорожі. Скільки нірок йому треба подолати для того, щоб поїсти моркви.

На місці нори знаходиться число, що вказує на клітку в яку веде дана нора.

			3	
	4			
5				
	2			

Технічні умови. У вхідному файлі **rabbits.dat** у першому рядку знаходиться три цілих числа: **N** ($1 < N < 100$) – розмір квадратної огорожі; **R**, **C** – рядок і стовпець умовної клітинки огорожі, де знаходиться кролик. У наступних **N** рядка записано по **N** чисел: 0 вказує на відсутність нори, інше число – на нору з переходом у клітку з таким номером. У вихідний файл **rabbits.sol** у випадку, коли кролик поїсть моркви, вивести “YES” у першому рядку та кількість нірок у другому. При відсутності шляху вивести в першому рядку “NO”, а в другому – номер останньої клітки, до якої може пройти кролик.

Вхідні дані

```
5 1 1
0 0 0 3 0
0 0 0 0 0
0 4 0 0 0
5 0 0 0 0
0 2 0 0 0
```

Вихідні дані

```
YES
3
```

Задача 26. «Головоломка»

(Заочна олімпіада, вища ліга, 2006-2007)

В країні Олімпія популярною є наступна головоломка. Задана матриця $M \times N$ натуральних чисел. За один крок вибирається деякий рядок з номером R , стовпчик з номером C , і до всіх чисел, які знаходяться на перетині рядка, що за номером не перевищує R , та стовпчика, що за номером не перевищує C , додається одиниця.

Потрібно за мінімальну кількість кроків зробити всі числа в матриці непарними. Напишіть програму, яка розв'яже описану головоломку.

Вхідні дані. В першому рядку файлу **matrix.dat** записані числа M і N – розміри матриці ($1 \leq M, N \leq 50$). В наступних M рядках по N натуральних чисел, що не перевищують 50 – елементи матриці. j - те число в $(i+1)$ - му рядку відповідає елементу на перетині i - го рядка та j - го стовпчика матриці.

Вихідні дані. В файл **matrix.sol** вивести мінімальну кількість операцій, яка необхідна для того, щоб всі числа матриці стали непарними.

Приклад вхідних і вихідних даних:

matrix.dat

3 3

1 4 8

3 4 2

2 1 7

matrix.sol

2

Примітка: Матрицю у прикладі можна перетворювати таким чином :

$$\begin{pmatrix} 1 & 4 & 8 \\ 3 & 4 & 2 \\ 2 & 1 & 7 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & 5 & 9 \\ 4 & 5 & 3 \\ 2 & 1 & 7 \end{pmatrix} \Rightarrow \begin{pmatrix} 3 & 5 & 9 \\ 5 & 5 & 3 \\ 3 & 1 & 7 \end{pmatrix}$$

Задача 27. «Кролик-холостяк»

(Заочна олімпіада, перша ліга, 2006-2007)

Фермер Вампіров, бізнес якого заснований на кроликах, зіткнувся з проблемою пошуку кролика, який не має пари. На його кролефермі кролики пронумеровані так, що кролики з однієї родини мають однакові номери, причому кількість кроликів у кожній родині парна. Ніякої системи у нумерації немає, лише відомо, що номер є натуральним числом, а також відомо, що кролик-холостяк – єдиний.

Вхідні дані: у текстовому файлі **unmarried.dat** через пропуск записані номери кроликів, які є натуральними числами і не перевищують $2 \cdot 10^9$.

Вихідні дані: у текстовий файл **unmarried.sol** записати єдине число – номер родини з непарною кількістю кроликів або номер кролика, у якого немає пари.

Приклад 1:

Вхідний файл: **unmarried.dat**

1 6 2 3 4 6 3 1 2 3 4 3 6

Вихідний файл: **unmarried.sol**

6

Приклад 2:

Вхідний файл: **unmarried.dat**

1 6 2 3 4 3 2 3 4 3 6

Вихідний файл: **unmarried.sol**

1

Задача 28. «Stack Attack (remix)»

(Заочна олімпіада, перша ліга, 2006-2007)

Діти фермерів зайнялись програмуванням і вирішили написати програму-гру "Stack Attack (remix)". Ігрове поле **Stack Attack** являє собою таблицю 6×8 (6 рядків, 8 стовпців). В кожній комірці можна розмістити квадратик одного з чотирьох кольорів: зелений, синій, червоний, жовтий.

Повністю заповнений квадратиками рядок записується в протокол і зникає з ігрового поля, після чого верхні рядки (якщо є) опускаються на один рядок вниз. Гра припиняється, коли в одному зі стовпців ігрового поля накопичується 6 квадратиків. Після закінчення гри всі 6 рядків ігрового поля також заносяться у протокол. За кожен повністю заповнений рядок нараховується 100 балів, за кожні три квадратики одного кольору підряд у заповненому рядку нараховується додатковий бонус у розмірі 23 балів (при цьому кожен квадратик можна рахувати лише один раз). Допоможіть дітям фермерів за даним протоколом гри визначити, скільки балів набрав гравець.

Вхідні дані: у кожному рядку текстового файлу **protokol.txt** записано 8 чисел через пропуск:

1 - позначає зелений квадратик, 2 - синій, 3 - червоний, 4 - жовтий, 0 - порожня комірка.

Вихідні дані: у перший рядок файлу **result.txt** вивести єдине число - кількість балів, набраних гравцем.

Приклад файлу protokol.txt:

```
1 2 2 2 2 3 3 1
3 0 1 4 3 4 3 2
3 0 2 2 2 4 3 1
2 0 3 3 2 4 1 0
1 0 1 0 0 2 2 0
4 0 2 0 0 0 0 0
0 0 1 0 0 0 0 0
```

Приклад файлу result.txt:

```
123
```

Задача 29. «Унікальна ціна»

(Заочна олімпіада, 2009-2010)

Фермер Дієтенко запровадив рекламну акцію, метою якої є розигриш головного призу – кролика. Головний приз дістанеться тому, хто запропонує

найменшу ціну на кролика і вона буде єдиною. Таку ціну назвемо – унікальною ціною. Допоможіть фермеру автоматизувати обробку отриманих пропозицій, тобто знайти унікальну ціну, яка подана у копійках.

Вхідні дані: у текстовому файлі `price.in` через пропуск записані натуральні числа, які не перевищують 50 000.

Вихідні дані: у текстовий файл `price.out` записати єдине число – унікальну ціну. Якщо такої ціни немає, то вивести «No»

Приклад 1:

Вхідний файл: price.dat

1 6 23 4 63 1 3 1 3 6

Вихідний файл: price.sol

4

Приклад 2:

Вхідний файл: price.dat

2 3 4 3 2 4

Вихідний файл: price.sol

No

Задача 30. «Свиноманія»

(Заочна олімпіада, 2009-2010)

В країні У сталася паніка. Свинячий грип вийшов з телевізійної віртуальної реальності і почав ходити по містах. Люди з переляку порозкупували всі ліки в аптеках (будь-які, головне ліки), понатягали маски та шарфи на обличчя, і перелякано чекали кінця Світу. В цей час свині на свинофермах не маючи телевізорів спокійно ставились до життя.

Також ставився до життя спокійно і програміст Вася, який вирішив піти в гості до своєї коханої Марічки. Але оскільки Вася ще неповнолітній, то мати заборонила йому йти на вулицю боячись, що дитина захворіє на страшну заморську хворобу. Ніякі вмовляння не допомогли програмісту Васі

здолати мамине переконання про небезпечність вірутального вірусу, і тоді Вася, вдався до геніального ходу.

Він склав карту міста. Карта являє собою прямокутник розмірами 1000x1000 з сторонами паралельними осям координат.

Південна вулиця має координати $(0,0) - (0,1000)$. Північна має координати $(1000,0) - (1000,1000)$

Він склав програму яка визначає чи є безпечний шлях з вулиці Північної (на якій він мешкає), до вулиці Південної (де мешкає його кохана Марічка) і продемонстрував це мамі. Вася впорався з завданням і зараз спілкується зі своєю коханою, а вам слабо?

Вхідні дані:

В першому рядку вхідного файлу `porckflu.in` міститься ціле число T – кількість наборів даних ($1 \leq T \leq 10$).

Далі йдуть самі набори. Перший рядок кожного набору містить одне число N – кількість уражених майданчиків ($0 \leq N \leq 1000$).

Кожен з наступних N рядків описує уражений майданчик (круг) і містить 3 числа: X_i, Y_i, R_i – координати центру і радіус ураження.

Вихідні дані:

Для кожного набору даних виведіть в окремий рядок вихідного файлу `porckflu.out` слово YES якщо, можна дістатись з північної до південної вулиці без ризику бути ураженим вірусом, та NO в іншому випадку.

Приклад вхідних даних:

```
3
500 500 499
0 0 999
1000 1000 200
```

Приклад вихідних даних.

```
YES
```

Задача 31. «Ділки»

(Заочна олімпіада, 2009-2010)

На боротьбу з примарою свинячого грипу ділки з фірми "A1H1 Ltd" вибили фінансування з державного бюджету. Тепер вони займаються дослідженнями і намагаються виявити вірус в різних матеріалах і речовинах. Вони розробили прилад, який за зразком проби переводить її у цифровий код – KOD - послідовність цілих позитивних чисел.

Тепер вони переймаються пошуком підпослідовності VIRUS - чисел які йдуть підряд у послідовності KOD і мають таку властивість: $VIRUS[1] > VIRUS[2] < VIRUS[3] > VIRUS[4] < VIRUS[5] \dots$

Дослідники впевнені, що чим більша послідовність VIRUS тим більше імовірність наявності вірусу свинячого грипу у пробній речовині.

Вхідні дані:

У вхідному файлі business.in перше число – N ($1 \leq N \leq 14000$). Далі N різних позитивних цілих чисел не більших за 30000.

Вихідні дані:

У вихідний файл business.out вивести єдине число – довжину найбільшої підпослідовності VIRUS в послідовності KOD

Приклад вхідних даних:

5 2 4 1 3 5

Приклад вихідних даних:

3

Задача 32. «Змійка»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Петрик з однокласниками полюбляє гратися у гру "Змійка". Сподіваюсь, вам також знайома ця гра: на прямокутному полі, розміром $N \times M$, переміщується змійка - неперервна ламана лінія шириною в 1 клітку,

яка може згинатися лише на 90 градусів. Змійка щоразу, проковтнувши здобич, "росте", і гра закінчується, коли "голова" змійки доторкнеться у довільному місці до "тіла". При русі паралельно своєму тілу змійка не може проходити у сусідніх клітинках.

Задане поточне розташування змійки. Потрібно вивести Yes, якщо голова змійки доторкнулась до тіла, тобто гра вже закінчилася, або No у протилежному випадку та порахувати довжину змійки.

Вхідні дані:

Вхідний файл z2.dat містить у першому рядку числа N і M, які задають розміри ігрового поля. ($1 \leq N, M \leq 100$)

У наступних N рядках містяться елементи масиву, кожен з яких дорівнює 1, якщо клітка належить змійці, або 0 - якщо клітка не належить змійці.

Вихідні дані:

У вихідний файл z2.sol вивести у перший рядок слово Yes, якщо гра програна, або слово No у протилежному випадку; у другий рядок вивести поточну довжину змійки.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
<pre> 10 10 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 </pre>	<pre> Yes 14 </pre>

0000000000	
0000000000	

Математика

Задача 1.

(I обласна олімпіада, 1987 р.)

Написати програму, яка знаходить методом поділу відрізка навпіл корінь рівняння $Ax + Bx + C = 0$, з точністю до E , якщо відомо, що корінь знаходиться на відрізку $[x_1, x_2]$. Перевірити роботу програми для $A=2,25$, $B=1,25$, $C=1,5$, $E=0,001$, $x_1=0$, $x_2=1$.

Задача 2.

(II обласна олімпіада, 1988 рік)

Скласти програму, за якою можна розрахувати значення функції $Y = ax+bx+cx+d$, не закладаючи x в регістри пам'яті. Вирішити задачу для $a=5$, $b=3$, $c=4$, $d=1$ і $x=1,75$, $x=2,5$.

Задача 3.

(III обласна олімпіада, 1988 рік)

Обчислюючи таблицю значень функції $y=f(x)$, $x=a, a+h, \dots, b$, визначити точки її локального мінімуму і максимуму. Крапку заданої шкали аргументів будемо називати локальним мінімумом, якщо $f(x) < f(x+h)$ і $f(x) < f(x-h)$; локальним максимумом, якщо $f(x) > f(x+h)$ і $f(x) > f(x-h)$. Функція $f(x) = \sin x + x/10$, $a=-2$, $b=5$, $h=0,2$.

Задача 4.

(IV обласна олімпіада, 1990 рік)

Форт має форму чотирикутника, і в його кутах і серединах сторін знаходяться майданчики, на яких можуть розташовуватися солдати. Складіть алгоритм, який допоміг би полковнику розставити n солдатів на цих майданчиках так, щоб уздовж кожної сторони знаходилося k солдатів або повідомив, що це зробити неможливо.

Задача 5.

(IV обласна олімпіада, 1990 рік)

Скласти програму отримання двійкового коду будь-якого цілого десяткового числа X ($0 \leq X \leq 255$).

Задача 6. «Біквadratне рівняння»

(VII обласна олімпіада, 1993 рік)

Побудувати алгоритм розв'язання біквadratного рівняння $ax^4+bx^2+c=0$.

Задача 7. «Свята»

(XIV обласна олімпіада, 2000 рік)

Уряд Ющенка занепокоєний великою кількістю неробочих днів в нашій державі. Дійсно, Новий Рік, Різдво, 8 Березня, 1 і 2 Травня і т.д. І це не рахуючи субот та неділь. Крім того, якщо свято випадає на суботу або неділю, то неробочим буде ще і найближчий робочий день, який йде слідом за святом. Наприклад, якщо 1 травня випаде на суботу, то неробочими будуть 1, 2, 3, 4 травня. У зв'язку з цим потрібно написати програму HOLIDAY.*, яка допомогла б проаналізувати процентне відношення робочих і неробочих днів в минулому, тепер і в майбутньому з метою врахування збитків від неробочих днів при складанні бюджету. Цікавим в даному плані є період з 1800 по 2200 рік.

Вхідні дані:

У першому рядку текстового ASCII файлу HOLIDAY.DAT через пропуск записані два числа у форматі ДДММРРРР, які означають період, що необхідно проаналізувати (наприклад, 06022000 означає 6 лютого 2000 року). Перше число означає початок періоду, друге - кінець. Обидва числа входять в період, що розглядається. Далі записані списки святкових днів для різних історичних етапів нашої держави у форматі: ДДММРРРР ДДММРРРР ДДММ ДДММ ДЦММ 0. Кінець файлу позначений числом 0

Примітки:

1. Гарантується, що період, який необхідно проаналізувати, повністю покривається одним або кількома історичними періодами, які не перетинаються.

2. При розрахунках потрібно використовувати сучасний (григоріанський) календар з врахуванням високосних років. (Високосним є рік, номер якого ділиться на 4. Вийняток є роки, номери яких діляться на 100 і не діляться на 400. Таким чином рік 2000 - високосний, а 1900 - ні).

3. Всі дати і періоди впорядковані в порядку зростання. Число періодів не перевищує 50, число святкових дат в періоді - 120. Вихідні дані: Текстовий ASCII-файл HOLIDAY.SOL повинен містити в одному рядку через пропуск два числа, перше з яких вказує на загальне число днів у періоді, друге - число неробочих днів у вказаному періоді.

Приклад вхідних і вихідних даних

HOLIDAY.DAT

01011999 31121999

01061946 01012000 0101 0201 0803 0105 0205 0905 0711 0811 0

0

HOLIDAY.SOL

365 112

Задача 8. «Біквдратне рівняння»

(XV обласна олімпіада, 2001 рік)

Складіть програму BIKVAD.PAS (або BAS чи CPP) знаходження дійсних коренів біквдратного рівняння $a*x^4+b*x^2+c=0$. Ваша програма повинна прочитати вхідні дані із текстового файла BIKVAD.DAT і вивести чотири корені або текст N0, якщо відповідний корінь відсутній, у текстовий файл BIKVAD.SOL.

Вхідні дані:

у чотирьох рядках файлу BIKVAD.DAT відповідно записані точність обчислення, виражена кількістю десяткових знаків k, коефіцієнти a,b,c.

Вихідні дані:

у чотири рядки файлу BIKVAD.SOL вивести значення коренів у порядку зростання або текст N0, якщо відповідний корінь відсутній (на першому місці ставити числа, а за ними текст).

Приклад вхідних і вихідних даних.

BIKVAD.DAT

4

1

-13

36

BIKVAD.SOL

-3.0000

-2.0000

2.0000

3.0000

Задача 9. «Вираз»

(XVIII обласна олімпіада, 2004 рік)

Задані дійсні числа c, d. Обчислити значення виразу з чотирма десятковими знаками

$$\left| \frac{\sin^3 |cx_1^3 + dx_2^2 - cd|}{\sqrt{(cx_1^3 + dx_2^2 - x_1)^2 + 3.14}} \right| + \operatorname{tg}(cx_1^3 + dx_2^2 - x_1),$$

де x_1 – більший, а x_2 – менший корені рівняння $x^2 - 3x - |cd| = 0$. Вхідні дані вводяться із вхідного стандартного потоку (клавіатура), а вихідні дані виводяться у стандартний вихідний потік (екран). Зайвих коментарів виводити не потрібно.

Задача 10.

(XVIII обласна олімпіада, 2004 рік)

Коли Максим Розуменко вчився у фінансовому коледжі, у нього були дуже погані стосунки з вчителем інформатики. Тому, на випускному екзамені вчитель дав йому індивідуальне завдання. Так як Максим був крутим хакером, він зміг скопіювати з сервера 10 тестів, на яких вчитель збирався тестувати його розв'язок. На жаль, на сервері не було відповідей на ці тести. І зараз Максиму якнайшвидше потрібно знайти відповіді.

Завдання.

Вчитель задав Максиму таке завдання:

"На проміжку $[A, B]$ необхідно знайти кількість натуральних чисел X таких, що:

- 1) $X, X+1, X+2, \dots, X+K-1$ належать $[A, B]$;
- 2) для кожного натурального Y з $[X, X+K-1]$ знайдеться натуральне $Z \triangleleft Y$ з $[X, X+K-1]$ таке, що у чисел Z і Y є хоча б один спільний дільник більший 1.

Крім того, знайдіть хоча б одне таке X (якщо воно існує)."

Вхідні данні.

Вам дається 10 вхідних файлів з іменами 1.IN, 2.IN, ..., 10.IN. В кожному з них записані три числа: K, A, B - по одному числу в рядку.

Вихідні дані.

Вам потрібно здати на перевірку 10 вихідних файлів з іменами 1.OUT, 2.OUT, ..., 10.OUT, де i .OUT - вихідний файл для i .IN ($i = 1, 2, \dots, 10$). Кожен з цих файлів повинен містити два рядки. В першому рядку повинно бути записана кількість чисел X , що задовільняють вищезаписані умови. В другому рядку повинно бути записане одне з таких X . У випадку, коли таких X не існує, файл повинен складатися з одного числа - 0.

Задача 11. «Про мікроби»

(XIX обласна олімпіада, 2005 рік)

Відомо, що в одному кубічному метрі повітря приміщення класу до початку занять міститься 2600 мікробів. З одного мікроба через кожні 30 хвилин при $t = +30^{\circ}\text{C}$ утворюється два. При $t = +20^{\circ}\text{C}$ розподіл мікробів сповільнюється у 2 рази, а при $t = +10^{\circ}\text{C}$ – у 20 разів. Об'єм класу $S \text{ м}^3$, у класі перебувають m учнів. Потрібно обчислити кількість мікробів, що припадають на одного учня через k годин після початку занять ($1 \leq k \leq 7$) при температурі t (t може мати значення 10, 20, 30).

Технічні вимоги: Із клавіатури вводяться (стандартний вхідний потік) S , m , k , t як чотири числа, що розділені пропуском. На екран виводиться (стандартний вихідний потік) єдине число – кількість бактерій, що припадають на одного учня.

Приклади вхідних та вихідних даних:

Вхідні дані:

120 10 4 30

120 15 2 20

100 20 5 10

Вихідні дані:

7987200

105300

21930

Задача 12. «Вивернута пам'ять»

(XIX обласна олімпіада, 2005 рік)

Після аварії в гіперпросторовому стрибку зореліт "ФАЭТОН" став дзеркально відбитим – ліве помінялося з правим. Серце у всіх астронавтів стало праворуч, а назва корабля перетворилась в "НОТЄАФ". Також «вивернутою» виявилась і пам'ять комп'ютера, що керує рухом корабля. Для виконання наступного гіперпросторового стрибка потрібно виправити

двійкове представлення програми в пам'яті комп'ютера. Для цього необхідно поміняти місцями біти в 16-бітних словах (виконати дзеркальне відображення): 0-й біт повинен стати 15-м, 1 – 14, 2 – 13 і т.д.

Приклад відображення: 0110001011101101 ® 1011011101000110

Формат вхідних даних: У першому рядку файла FAETON.DAT записане одне ціле число N ($0 < N$)

Формат вихідних даних: У вихідний файл NOTEAF.REZ записати N рядків, по одному числу в рядку – десяткове представлення слів програми після дзеркального відображення.

Приклад файла FAETON.DAT:

3

1

0

32767

Приклад файла NOTEAF.REZ:

32768

0

65534

Задача 13. «Перевірте рівність»

(XXII обласна олімпіада, 2008 рік)

Задані цілі числа X, Y, P, Q ($-10^{100} \leq X, Y, P, Q \leq 10^{100}$). Потрібно перевірити рівність $X^Y = P^Q$. Нагадаємо, що a^b визначається таким чином:

при $b > 0$, $a^b = a * a * \dots * a$ (b множників)

при $b = 0$, $a \neq 0$ $a^b = 1$

при $b < 0$, $a \neq 0$ $a^b = 1/a^{-b}$

для інших комбінацій a і b значення a^b не визначене.

Формат вхідних даних

У вхідному файлі записані числа X, Y, P, Q , кожне в окремому рядку.

Формат вихідних даних

Виведіть слово `correct`, якщо дана рівність для отриманих вхідних даних виконується, або `incorrect`, якщо рівність не виконується, або хоч би один із ступенів не визначений.

Приклади

g.in	g.out
2	correct
4	
4	
2	
2	incorrect
3	
3	
2	

Задача 14.

(XXIII обласна олімпіада, 2009 рік)

Петрик та Сергій під час зимових канікул часто грали у гру «Великий добуток». Суть цієї гри така: множать ціле число P на одне з чисел від 2 до 9. Петрик завжди починає з $P = 1$, виконує множення, потім отримане число множить Сергійко, потім знову Петрик і т.д. Перед початком гри вони вибирають випадкове число N , і переможцем виходить той, хто перший отримає $P \geq N$. Попробуйте визначити, хто із хлопчиків виграє при заданому N , якщо обоє грають оптимально правильно.

Задача 15. «Triangle»

(XXIV обласна олімпіада, 2010 рік)

Трикутне число — це число кружечків, які можуть бути розставлені у формі рівностороннього трикутника:



$$T_2=3 \quad T_3=6$$

Послідовність трикутних чисел T_n для $n = 0, 1, 2, \dots$ починається так: 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55,...

Напишіть програму, яка знаходить N -е трикутне число.

Задача 16. Одиничні дроби

(Заочна олімпіада, 2001-2002)

Потреба в дробах виникла ще в давні часи. Перший дріб, який використали, був, очевидно, "половина" ($1/2$). Потім з'явилися дроби з одиницею в чисельнику, так звані "одиничні" дроби: $1/3, 1/4, 1/5, \dots$ Якщо розрахунки призводили до появи інших дробів, то їх замінювали сумою різних одиничних дробів. Наприклад, $2/5 = 1/3 + 1/15$.

Напишіть програму для запису звичайного дроби (≤ 1) у вигляді суми одиничних дробів.

Вхідні дані знаходяться у файлі **INPUT.DAT**, в єдиному рядку якого через пропуск записано два числа: чисельник і знаменник дроби, що не перевищують 1000.

Вихідні дані: вивести у файл **OUTPUT.DAT** результат у вигляді послідовності знаменників одиничних дробів (через пропуск).

Приклад:

INPUT.DAT

2 5

OUTPUT.DAT

3 15

Задача 17.

(Заочна олімпіада, 2002-2003)

Дано два цілих числа M і N .

Обчисліть суму всіх цілих чисел від меншого до більшого з даних чисел (включно).

Вхідні дані: перший рядок - число M ($-2000000000 < M < 2000000000$);
другий рядок - число N ($-2000000000 < N < 2000000000$).

Вихідні дані: відповідне число.

Приклад вхідних даних:

1
2

Приклад вихідних даних:

3

Задача 18. «Десятковий дріб»

(Заочна олімпіада, 2002-2003)

Дано дріб m/n , де m та n цілі числа, $0 \leq m \leq 1000000$, $1 \leq n \leq 1000000$.

Вивести точне значення дробу в десятковому вигляді з виділенням періоду.

Наприклад:

$$1/3 = 0.(3)$$

$$15/14 = 1.0(714285)$$

Вхідні дані:

В першій стрічці вхідного потоку міститься два числа m та n розділені одним або кількома пропусками.

Вихідні дані:

Вивести в першу стрічку вихідного потоку значення дробу в десятковому вигляді.

Не слід виводити десяткову крапку чи дужки періоду без необхідності.

Приклади даних:

Вхідні дані

Вихідні дані

1 1

1

25 10

2.5

100 6

16.(6)

1 7

0.(142857)

15 14

1.0(714285)

Задача 19. «Степінь по модулю»

(Заочна олімпіада, 2002-2003)

Обчисліть значення виразу: залишок від ділення на m x в степені y .

$z = x^y \bmod m$, де x, y, m -- натуральні числа, що не перевищують 1000000000.

Вхідні дані:

В першій стрічці вхідного потоку містяться числа x, y, m розділені одним або більше пропусками.

Вихідні дані:

В першу стрічку вихідного потоку вивести число z -- значення виразу.

Приклади даних:

Вхідні дані	Вихідні дані
2 7 1000	128
2 7 10	8
10 9 11	10
3 4 4	1

Задача 20. «Лінивець»

(Заочна олімпіада, перша ліга, 2005-2006)

"Не відкладай на завтра... те, що можна відкласти на післязавтра!". Так перефразував народну мудрість один лінивець, перелазячи в своїх джунглях з гілки на гілку. А щоб не забути, на коли вікладає свої справи, він придбав комп'ютер. Тепер йому дуже потрібна програма, яка б правильно вираховувала число i місяць, що настануть післязавтра, якщо відомі сьогоднішнє число i місяць. 31 грудня 2006 року Лінивець планує поміняти комп'ютер на більш досконалий, тому його влаштує програма, яка працює правильно з датами в межах 2005-2006 років.

Вхідні дані: з стандартного вхідного потоку програма отримує два цілих числа C - сьогоднішнє число ($0 < C < 32$) і M - сьогоднішній місяць ($0 < M < 13$).

Вихідні дані: в два рядки стандартного вихідного потоку програма виводить два цілих числа - післязавтрашні число і місяць.

Приклад вхідних даних:

20 7

Приклад вихідних даних:

22

7

Задача 21.

(Заочна олімпіада, перша ліга, 2005-2006)

Двоє товаришів домовляються про зустріч через певний час. Один з них має лише годинник з циферблатом, на якому години позначаються числами від 1 до 12. Другий має лише секундомір, на якому облік часу ведеться у секундах і може виражатися невід'ємним цілим числом. Коли годинник показував час **H** годин **M** хвилин **S** секунд, був увімкнений секундомір. З'ясувати, який час буде на годиннику, коли за секундоміром пройде **T** секунд.

Вхідні дані: В першому рядку текстового файлу **clock.dat** записані три цілих числа **H**, **M** та **S**, розділені пропусками – показання годинника на початку відліку ($0 < H < 13$, $0 \leq M < 60$, $0 \leq S < 60$). В другому рядку – єдине число, час на секундомірі **T** ($0 \leq T \leq 1\,000\,000\,000$).

Вихідні дані: В єдиному рядку текстового файлу **clock.sol** записані три числа, розділені пропусками – кількість годин, хвилин та секунд на годиннику через **T** секунд після початку відліку.

Приклад 1:

clock.dat	clock.sol
10 15 20	10 15 55
35	

Приклад 2:

clock.dat	clock.sol
10 15 20 350	10 21 10

Задача 22. «Прямокутник»

(Заочна олімпіада, 2008-2009)

Назвемо "довжиною" прямокутника, сторони якого паралельні осям координат, його розмір вздовж осі ОХ, а "шириною" - розмір вздовж осі ОУ. Зі стандартного вхідного потоку вводяться чотири цілі числа X_1, Y_1, X_2, Y_2 , відокремлені пропусками, модуль жодного з яких не перевищує 1000000000 - координати двох протилежних вершин прямокутника. Визначити його довжину та ширину і вивести у стандартний вихідний потік у такому форматі:

LEN:<довжина>

WID:<ширина>

Приклад вхідних даних:

2 4 -3 90

Приклад вхідних даних:

LEN:5

WID:86

Задача 23.

(Заочна олімпіада, 2008-2009)

Для забезпечення кормами кроликів фермер Дістенко придбав зернову культуру у паспорті господарської придатності якої були записані такі дані:

N – зернин мають масу M г, чистота насіння становить $K\%$, схожість $C\%$ та вказано, що на одному гектарі має рости P паростків даної культури.

Написавши листа фермеру Вампірову із проханням допомогти встановити необхідну кількість зерна для засівання N гектарів площі при

таких даних: $N=1000$, $M=40$, $K=97$, $C=93$, $P=6000000$, $H=100$ він отримав таку відповідь: 26604.589 кг. Напишіть програму, яка автоматизує процес обчислення.

Технічні умови:

Вхідні дані: у єдиному рядку текстового файлу `ekonom.in` через пропуск записані натуральні числа: N , M , K , C , P , H

Вихідні дані: у єдиний рядок текстового файлу `ekonom.out` записати знайдену масу зерна, виражену у кілограмах з точністю до грама.

Приклад вхідних даних:

1000 40 97 93 6000000 100

Приклад вихідних даних:

26604.589

Задача 24.

(Заочна олімпіада, 2008-2009)

У фермера Дієтенка сталася прикра недоречність: вийшов з ладу годинник, який так довго йому слугував. Тепер час він міг визначати тільки за метрономом. Робота фермера вимагає точнішого виміру часу. Метроном він може налаштувати на K ударів за T секунд. Як знаючи кількість N ударів визначити час у годинах, хвилинах та секундах.

Вхідні дані: у єдиному рядку текстового файлу `time.in` записані через пропуск натуральні числа N , T , K .

Вихідні дані: у єдиний рядок текстового файлу `time.out` записати результат у вигляді:

h:годинит:xвилини:s:секунди

Приклад вхідних даних:

10013 1 2

Приклад вихідних даних:

h:1m:23s:26

Приклад вхідних даних:

10013 2 1

Приклад вихідних даних:

h:5m:33s:46

Задача 25. «Проміле»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Дмитрик захопився точними розрахунками. Він добре вивчив проценти (1/100 частина) і був вражений новою інформацією: 1/1000 частина чогонебудь називається проміле і позначається ‰. Саме така точність йому була потрібна для розрахунків.

Завдання. В файлі promile.in в кількох рядках через пропуски записано проміле. Необхідно додати всі частини і визначити цілу і дробову частини. Результат записати через кому в файл promile.out.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
123 678 399 456	1,656

Задача 26. «Куб»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Дерев'яний куб зі стороною $a > 1$ пофарбували та розпиляли на кубики зі стороною 1. Визначити кількість кубиків, які матимуть 3 пофарбовані грані, дві пофарбовані грані, одну та жодної.

Обмеження:

Сторона куба є натуральним числом, що не перевищує 2010.

Вхідні дані:

Із стандартного потоку прочитати число a , що є стороною куба.

Вихідні дані:

У єдиний рядок стандартного потоку через пропуск записати кількість кубиків, які матимуть три, дві, одну та жодної пофарбованих граней.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
3	8 12 6 1

Задача 27. «Сума коефіцієнтів»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Знайдіть суму коефіцієнтів при непарних степеня x многочлена $(x^5+x-1)^N$.

Обмеження:

N є натуральним числом, що не перевищує 2010.

Вхідні дані:

Із стандартного потоку прочитати число N .

Вихідні дані:

У єдиний рядок стандартного потоку записати суму коефіцієнтів при непарних степеня x вказаного многочлена.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
2	-4

Математичне моделювання

Задача 1.

(II обласна олімпіада, 1988 рік)

Взуттєва фабрика щорічно збільшує виробництво взуття на $p\%$. Скласти програму, за якою можна розрахувати через скільки років виробництво взуття подвоїться. Вирішити завдання для $p = 7\%, 13\%, 19\%$.

Задача 2. «Фермер»

(XII обласна олімпіада, 1998 рік)

Фермер зібрав гарний урожай і перед ним постала проблема якомога вигідніше його продати. Фермеру, використовуючи можливості Internet, вдалось отримати повну інформацію про ціни на ринках в сусідніх селах та ціни на пальне. Треба написати програму FARMER.*, що пропонує село, на ринок якого найбільш вигідно відвезти продукцію. У фермера достатньо пального щоб дістатись будь-якого з ринків, але, можливо, потрібно буде купити пальне на зворотній шлях.

Задача 3. «Перекладачі»

(XVI обласна олімпіада, 2002 рік)

В складних умовах ринкової економіки директор фірми "POLYGLOT", що спеціалізується на перекладі літератури з інформатики з N різних мов, вирішив зменшити витрати своєї фірми. Для цього він вирішив звільнити з роботи деяких перекладачів.

Напишіть програму, яка визначить працівників, яких потрібно залишити на роботі за таких умов:

- 1) Кожен перекладач знає дві мови (M_1 M_2), та отримує заробітну платню C .
- 2) Бюро повинно перекладати тексти з будь-якої мови на будь-яку іншу.

3) Сумарна заробітна платня повинна бути мінімальною

Вхідні дані:

у першому рядку два числа, N та K . ($2 \leq N \leq 100$, $K \leq 30000$) у наступних K рядках міститься по три числа: $M1i$ $M2i$ Ci - мови які знає i -й перекладач, та його заробітна платня. ($1 \leq Ci \leq 100$).

Вихідні дані:

у першій рядок вивести число P - кількість працівників, яких потрібно залишити на роботі. у другий рядок вивести P чисел - номери перекладачів, яких потрібно залишити на роботі.

Приклад:

6 7

1 2 4

1 6 3

2 3 5

2 5 4

2 6 2

3 4 4

4 5 1

Відповідь:

5

2 5 4 7 6

Задача 4.

(XXIII обласна олімпіада, 2009 рік)

N народних депутатів збираються зібратися на благодійний аукціон. i -й депутат приходить в момент часу $T[i]$ і має можливість витратити на цьому аукціоні $P[i]$ не наших грошей. Двері приміщення аукціону мають $K+1$ ступенів відкритості і позначаються цілими числами з інтервалу $[0, K]$. Ступінь відкритості дверей може змінюватися на одиницю за одиницю часу, тобто двері можуть відкритися на одиницю, закритися на одиницю або

залишитися в попередньому стані. В початковий момент часу двері зачинені ($K=0$). i -й депутат заїде на аукціон лише тоді, коли двері відчинені спеціально для нього (така собі людська слабкість), тобто коли ступінь відкритості дверей відповідає його повноті $S[i]$. Якщо в момент, коли депутат під'їде до будинку аукціону, ступінь відкритості дверей не відповідає його повноті, він їде і більше не повертається. Аукціон працює в інтервалі часу $[0, T]$. Організатори хочуть зібрати депутатів з максимальним сумарним багатством відкриваючи та закриваючи двері відповідним чином. На яку суму виторгу вони можуть розраховувати.

Задача 5. «Salon»

(XXIV обласна олімпіада, 2010 рік)

В салон краси прийшло n дівчат. Кожна з них повинна відвідати перукаря і косметолога. У кожного вона проводить по m годин. За який найменший час k співробітників салону краси зможуть обслужити всіх дівчат, якщо кожен співробітник може виконувати функції як косметолога, так і перукаря?

Задача 6.

(Заочна олімпіада, 2003-2004)

Над послідовністю, яка складається з N чисел натурального ряду, виконують такі дії:

- 1) викреслюють послідовно кожний M -й елемент в послідовності;
- 2) якщо в послідовності залишилось більше або рівно M елементів переходимо до дії 1.

Яке число буде викреслено останнім в результаті проведених дій. Якщо число не існує, відповідь: 0.

Вхідні дані: в одному рядку через пропуск натуральні числа M і N .

Вихідні дані: відповідне число.

Технічні обмеження: M і N не перевищують 2000000000.

Приклад вхідних даних:

100 100

Приклад вихідних даних:

100

Задача 7. «Бібліотека кошового писаря»

(Заочна олімпіада, 2003-2004)

В бібліотеці кошового писаря є зібрання з N книжок. ($N \leq 1000$). Він вирішив упорядкувати їх у книжкову шафу. Шафа має ширину $W \leq 2000000000$ і писар може обирати висоти книжкових полиць такі, які лише захоче. Але йому необхідно підібрати їх так, щоб загальна висота шафи була найменш можливою. Порядок книжок змінювати не можна! (Товщини та висоти книжок - цілі додатні числа, що не перевищують 2000000. Ширина шафи гарантовано не менша за товщину найтовстішої книги)

Вхідні дані:

У першому рядку числа N та W .

Другий рядок містить N цілих чисел - висоти книжок.

Третій містить також N цілих чисел - товщини книжок.

Приклад:

3 5

2 4 4

2 2 2

Вихідні дані:

Ціле число - мінімально можлива висота шафи.

Вихідні дані до прикладу:

6

Задача 8. «Добровільна пожертва»

(Заочна олімпіада, перша ліга, 2005-2006)

У загадковій країні Труляля, після чергової кольорової революції, у боротьбі із корупцією ввели нові грошові купюри номіналом **1, 3, 9, 27, 81...** Поки кольорові воюють із міфічними корупціонерами, шустрый бармен свої

«старі» звички переніс на новий лад, назвавши чайові добровільною пожертвою. Ваша задача допомогти мандрівнику мінімізувати витрати, тобто заплатити за чеком і, не порушуючи спокій у країні, видати якнайменшу добровільну пожертву бармену. Обмінні пункти у країні Труляля видають тільки по одній купюрі одного номіналу і у кишені мандрівника всіх нових купюр рівно по одній. Інфляційні процеси ще дають можливість «відпочити» на суму в межах одного мільярда.

Вхідні дані: В перший рядок текстового файлу **tugrik.in** записана сума, яку потрібно заплатити, згідно чеку.

Вихідні дані: В єдиний рядок текстового файлу **tugrik.sol** записати суму, яку потрібно віддати бармену, врахувавши добровільну пожертву.

Приклад вхідних і вихідних файлів:

tugrik.in	tugrik.sol
79	81

Задача 9. «Аеропорт»

(Заочна олімпіада, вища ліга, 2006-2007)

Компанія "Хмельницькі авіалінії" бажає здійснювати пасажирські перевезення на низці рейсів виду Хмельницький -- деяке місто -- Хмельницький. Виліт літака на кожен з рейсів здійснюється щоденно. Для кожного рейсу відома необхідна година виходу літака на рейс та година звільнення літака з рейсу після його повернення в аеропорт міста Хмельницького. Наприклад, Хмельницький -- Одеса -- Хмельницький 15:35 - 18:20, Хмельницький -- Стамбул -- Хмельницький 12:50 -- 17:45. Якщо проміжки часу, необхідні для обслуговування рейсів не перетинаються, то ці рейси може обслуговувати один літак.

Задача. За даним списком рейсів, що визначені часом початку та закінчення обслуговування рейсу, визначити найменшу кількість літаків, яка зможе обслуговувати всі ці рейси. Кількість рейсів не перевищує 1000.

Вхідні дані. В першій стрічці вхідного потоку міститься одне число -- кількість рейсів. В наступних стрічках міститься опис рейсів, по одному в кожній стрічці, у вигляді: час початку - час завершення. Наприклад, 13:20 - 18:10. Значення годин та хвилин завжди містять дві цифри. Значення часу та знак "-" можуть бути відокремлені деякою кількістю пропусків. Вхідні дані гарантовано містять коректне значення годин та хвилин.

Вихідні дані. В першу стрічку вихідного потоку вивести єдине число -- найменшу кількість літаків, що можуть обслуговувати всі дані рейси.

Приклад вхідних даних:

5
03:50 - 09:10
05:15 - 08:45
08:55 - 12:20
10:15 - 15:35
23:10 - 03:30

Приклад вихідних даних, що відповідають вхідним:

2

Задача 10. «Метеоритний дощ»

(Заочна олімпіада, вища ліга, 2006-2007)

Вчені астрономи, повідомили уряд, що рівно через T секунд на планеті Олімпіада почнеться метеоритний дощ. Уряд планети терміново почав евакуацію людей і важливого обладнання - телескопічних установок.

Для захисту телескопу його потрібно транспортувати у спеціально відведене місце - сховище. В одному сховищі може вміститися лише один телескоп.

Необхідно розрахувати максимальну кількість телескопів, які можна встигнути заховати до початку метеоритного дощу.

Вхідні дані (файл METEOR.DAT):

У першому рядку число T - кількість секунд, через яку метеорити впадуть на планету.

У другому рядку число N - кількість телескопів.

У третьому рядку знаходиться $2*N$ чисел - координати телескопів (x , y).

У четвертому рядку число M - кількість сховищ.

У п'ятому рядку знаходиться $2*M$ чисел - координати сховищ (x , y).

У шостому рядку число V - швидкість транспортування телескопів.

Обмеження: $0 < T < 1000000$; $0 < N < 100$; $0 < M < 100$; $-30000 \leq x \leq 30000$; $-30000 \leq y \leq 30000$; $0 < V < 1000$

T , N , M , x , y - цілі.

Вихідні дані (файл **METEOR.SOL**):

У єдиний рядок файлу записати максимальну кількість телескопів, що можуть бути сховані до початку падіння метеоритів.

Приклад:

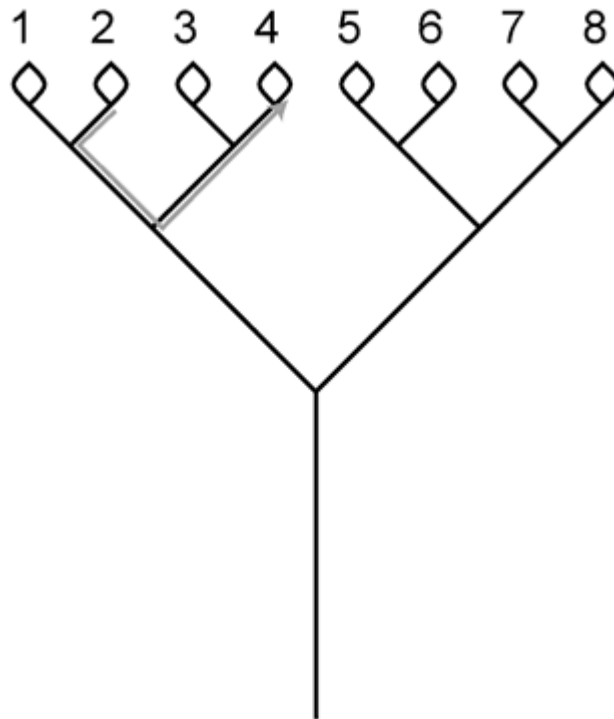
METEOR.DAT	METEOR.SOL
10	2
4	
1 7 3 7 5 7 100	
100	
2	
2 7 4 7	
0.1	

Задача 11. «Двійкове дерево»

(Заочна олімпіада, вища ліга, 2006-2007)

Двійкове дерево розміром n складається з стовбура довжиною 2^n . Від стовбура ростуть дві гілки довжиною 2^{n-1} кожна. На кінці кожної такої гілки ростуть менші дві гілочки довжиною 2^{n-2} . І так далі. Довжина найменших гілочок складає 2^0 . На кінці кожної найменшої гілочки росте листочок. Листочки пронумеровані числами від 1 до 2^n таким чином, що номери всіх

листочків відрізняються та номери листочків будь-якої гілки дерева йдуть підряд без проміжків. На листочку i сидить гусениця. Їй не сподобався смак цього листочка і вона хоче переповзти на листочок з номером j . Який найменший шлях має здолати гусениця по гілках двійкового дерева, щоб досягти листочка j ?



Початкові дані:

В першому рядку вхідного потоку містяться три числа: n , i та j розділені пропусками. $1 \leq n \leq 30$, $1 \leq i, j \leq 2^n$.

Результат:

В перший рядок вихідного потоку вивести одне число – довжину шляху від листочка i до j .

Приклад початкових даних:

Дані, що відповідають малюнку наведеному вище:

3 2 4

Приклад результату:

Результат, що відповідає даним наведеним вище:

6

Задача 12. «Game»

(Заочна олімпіада, 2009-2010)

Вовочка любить на уроках математики грати різні інтелектуальні ігри – не байдикувати ж зовсім. Недавно він захопився такою простою грою. Спочатку для гри вибирає деяке натуральне число N і малює на папері кружечок. Потім малює ще один і з'єднує його з першим лінією. Після цього малює наступний кружечок і з'єднує його з одним із попередніх. Так він робить до тих пір, поки на листку не буде рівно N кружечків. Свої кружечки Вовочка нумерує в довільному порядку числами від 1 до N . Складіть програму, що визначить скільки різних малюнків може зробити Вовочка для вибраного N . Малюнки вважаються різними, якщо існують такі кружечки з номерами i та j , що на одному малюнку вони з'єднанні, а на іншому – ні.

Формат вхідних даних. У першому рядку вхідного файлу `game.in` міститься K ($0 < K \leq 10$) – кількість тестів. Наступні K рядків містять натуральне число N ($0 < N \leq 100$) для кожного тесту.

Формат вихідних даних. Для кожного тесту у вихідний файл `game.out` вивести залишок від ділення кількості різних малюнків на число 2000000011.

Приклад вхідних та вихідних даних.

game.in

3

1

2

3

game.out

1

1

3

Обробка рядків

Задача 1.

(I обласна олімпіада, 1987 р.)

Дана рядкова величина ТЕКСТ. Написати алгоритм, який визначає, яка літера в цьому тексті зустрічається частіше всього. Виходити з припущення, що така літера одна.

Задача 2.

(II обласна олімпіада, 1988 рік)

Скласти алгоритм, що підраховує число тих слів у фразі, в яких у другій позиції знаходиться задана літера. Слова розділені одним або декількома пропусками, інших знаків пунктуації немає.

Задача 3.

(III обласна олімпіада, 1989 рік)

У рядковій змінній ТЕКСТ записано текст-послідовність слів; слово – послідовність літер. Два сусідніх слова поділяються крапкою, комою або пробілом.

Скласти алгоритм, переписували тексти в літерних таблицю СТОРІНКА [1: M] так, щоб у кожному її елементі-рядку було не більше N символів. Перенесення слів з рядка на рядок не допускається; довжина слів в тексті не перевершує N. Число рядків, заповнених текстом, має бути мінімальний. Точки та коми повинні зберегтися, але вони не повинні з'являтися на початку нового рядка.

Задача 4.

(IV обласна олімпіада, 1990 рік)

У деякому арифметичному виразі видалили всі символи, крім дужок. Скласти алгоритм, який визначає за отриманою послідовності дужок, чи правильно вони були розставлені.

Задача 5. «Робочі дні»

(V обласна олімпіада, 1991 рік)

Місяцю з 31-го дня відповідає лінійна таблиця елементів, в якій робочий день позначено 1, а неробочий – 0. Скласти алгоритм, який підраховує:

- а) кількість тижнів в місяці;
- б) кількість робочих днів в кожному з тижнів місяця.

Примітка: Тижнем будемо називати довільну не пусту множину одиниць між двома послідовними нулями або обмежену нулем праворуч (якщо місяць починається з робочого дня), або обмежену нулем ліворуч (якщо місяць не закінчується робочим днем).

Задача 6. «Фіксована літера»

(VI обласна олімпіада, 1992 рік)

Написати алгоритм, який підраховує кількість тих слів в реченні з N символів, в яких на другій позиції знаходиться задана літера. Слова відокремлені пропусками. Інших розділових знаків немає.

Задача 7. «Дужки»

(VII обласна олімпіада, 1993 рік)

Проаналізувати заданий текст з метою виявлення помилок у використанні дужок. Можливі три типи помилок:

- а) невідповідність дужок (і) по кількості;
- б) закриваюча дужка розміщена раніше відкриваючої;
- в) відсутній зміст між дужками.

Результатом роботи програми повинно бути повідомлення про типи допущених помилок та їх місце в тексті (якщо це можливо).

Задача 8. «Визначення сторінок»

(XI обласна олімпіада, 1997 рік)

При друкуванні великих документів може виникнути потреба друкувати не весь документ, а тільки деякі його сторінки. Серед аргументів програми друку є рядок з послідовністю номерів сторінок. Потрібно надрукувати не окремі сторінки, а діапазони сторінок i , можливо, вказувати початок і кінець діапазонів, а не послідовні числа.

Завдання: Напишіть програму, яка буде перетворювати списки сторінок у відповідну послідовність номерів сторінок.

Вхідні дані: Вхідний файл PRINT.DAT містить один рядок, який має таку структуру: сторінка-1, сторінка-2, сторінка-3, ..., сторінка – N.

Сторінка – i – або номер сторінки, або діапазон у вигляді початок–кінець (початок \leq кінець).

Сторінки та діапазони перераховані в зростаючому порядку і не перетинаються. Діапазон включає початкову та кінцеву сторінки. Номери сторінок – числа від 1 до 1000000. $1 \leq N \leq 1000000$.

Вихідні дані: Результат треба вивести до файлу PRINT.SOL у вигляді сторінка-1, сторінка-2, сторінка-3, ..., сторінка – M без пропусків.

Задача 9. «Форматування тексту»

(XIII обласна олімпіада, 1999 рік)

Дано текст. Виконати форматування тексту по ширині за заданою лівою і правою границею і абзацному відступу. Всі символи тексту з кодами 0..32 вважати пробільними, решту непробільними. Вирази в тексті виду "<n3 n1 n2>" означають початок абзацу з лівою границею $n1$, правою границею $n2$ і абзацним відступом $n3$ і повинні бути виключені із відформатованого тексту. Вважати, що текст завжди починається з такого виразу. Вважати, що вирази завжди коректні ($n2 > n1$, $n2 > n3$, $n1 > 0$, $n2 > 0$, $n3 > 0$). Вважати, що в тексті немає слів довжиною більше $(n2 - n3) / 2 - 1$. Відформатований текст із

пробільних символів повинен містити тільки пробіли (код 32) і переводи стрічки (пара символів з кодами 13, 10 якраз в такому порядку).

Задача 10. «Повідомлення»

(XIV обласна олімпіада, 2000 рік)

Із ліцею м.Шепетівки в ліцей "Антей" м.Кам'янець-Подільського з допомогою двох кур'єрів було відправлене повідомлення, яке складається із символів 0 і 1. Ради шутки кожний із кур'єрів проробив декілька разів таке перетворення повідомлення: будь-яку частину, що містить дві одиниці записав у зворотному порядку (перевернув). Наприклад, в повідомленні 11010100 кур'єр міг перевернути підрядок, який складається із символів із 2 по 5 позиції, і тоді отримувалось повідомлення 10101100. Отримавши два повідомлення в м. Кам'янець-Подільському ліцеїсти "Антею" вирішили перевірити їх еквівалентність, тобто чи можна отримати одне з іншого з допомогою описаних перетворень. Потрібно написати програму REPORT.*, яка визначає еквівалентність повідомлень і у випадку еквівалентності повідомлень знаходить хоча б один спосіб перетворення одного повідомлення в інше.

Вхідні дані:

У текстовому ASCII-файлі REPORT.DAT у двох рядках записані отримані в м.Кам'янці-Подільському повідомлення. Їх довжина не перевищує 100 символів.

Вихідні дані:

Якщо повідомлення не еквівалентні, то текстовий ASCII-файл REPORT.SOL повинен містити тільки один рядок NO. Якщо повідомлення еквівалентні, то в перший рядок текстового ASCII-файла REPORT.SOL потрібно вивести слово YES, в наступні рядки потрібно вивести послідовність перетворень першого повідомлення в друге. Кожне перетворення записується в окремому рядку у вигляді пари чисел i, j ,

розділених пропуском, що означає перевертання підрядка символів від і-го по j-ий номер.

Приклад вхідних і вихідних даних

REPORT.DAT	REPORT.SOL
100011100	YES
001011001	6 9
	3 8
	1 5

Задача 11. «Коректор»

(XV обласна олімпіада, 2001 рік)

При роботі з текстовим редактором використовують операції вставки символу в рядок, знищення символу та заміни символу. Назвемо ці операції редагування елементарними. Ви повинні скласти програму CORECT.PAS (або BAS чи CPP), яка прочитає з текстового файла CORECT.DAT два символні рядки S1 та S2, підрахує найменшу кількість елементарних операцій редагування, які необхідно виконати, щоб рядок

S1 перетворити в рядок S2 і результат виведе в єдиний рядок текстового файла CORECT.SOL.

Вхідні дані:

у першому рядку файлу CORECT.DAT записаний символний рядок S1, у другому рядку - символний рядок S2.

Вихідні дані:

у текстовий файл CORECT.SOL слід записати єдине число - найменшу кількість елементарних операцій редагування.

Приклад вхідних і вихідних даних.

CORECT.DAT	CORECT.SOL
принтер	
спринтер	

1

Задача 12. «Word»

(XX обласна олімпіада, 2006 рік)

Із стандартного вхідного потоку вводиться речення з крапкою в кінці. Визначити і вивести у стандартний вихідний потік слово, що трапляється у реченні найчастіше.

Задача 13. «Шпигуноманія»

(XX обласна олімпіада, 2006 рік)

Агент Джеймс Бонд передає свої повідомлення в Центр за допомогою електронної пошти. Лист складається з N рядків ($1 \leq N \leq 50000$), довжина яких не перевищує 255 символів. Для того, щоб затрудняти розшифровку повідомлення в лист поміщається «сміття», яке складається з рядків, що повторюються. Кожний рядок, що відноситься до «сміття» повторюється парну кількість разів, і лише один рядок зустрічається в листі тільки один раз (це і є власне повідомлення агента). Вам необхідно написати програму, яка із заданого тексту листа визначить текст повідомлення Джеймса Бонда.

Вхідні дані:

Вхідний потік містить набір рядків листа. Ознакою закінчення листа є символ «#», який стоїть в окремому рядку (даний рядок до тексту листа не відноситься).

Вихідні дані:

У вихідний потік необхідно вивести знайдений рядок повідомлення.

Наприклад:

Input	Output
Куй залізо, не відходячи від каси Алло шеф, це я - Льолік Битиму акуратно, але сильно Битиму акуратно, але сильно Куй залізо, не відходячи від каси #	Алло шеф, це я - Льолік

Задача 14. «Smile»

(XXIV обласна олімпіада, 2010 рік)

Напишіть програму, яка підрахує кількість смайликів у заданому тексті. Смайликом будемо вважати послідовність символів, які задовольняють умови: першим символом є або ; (крапка с комою) або : (двокрапка) рівно один раз далі може йти символ – (мінус) скільки завгодно разів (у тому числі символ мінус може йти нуль разів) у кінці обов'язково йде деяка кількість (не менше однієї) однакових дужок із такого набору: (,), [,]. всередині смайлика не може зустрічатися ніяких інших символів.

Наприклад, нижче наведені послідовності є смайликами:

:)

;-----[[][[[]

в той час як ці послідовності смайликами не є (хоча деякі з них містять смайлики):

:-)]

;--

-)

::-(

:-()

В цій задачі потрібно буде підрахувати кількість смайликів, які містяться у даному тексті.

Задача 15. «Анаграма»

(Заочна олімпіада, 2000-2001)

Написати програму, що визначає, чи можна з однієї точки рядка отримати іншу за допомогою перестановки літер і цифр, перетворень великих літер в маленькі і навпаки, довільного видалення і додавання інших символів.

Задача 18. «Найбільше число»

(Заочна олімпіада, 2002-2003)

Дано рядок символів, в якому обов'язково містяться цифри. Напишіть програму, яка знаходить в даному рядку максимальне число.

Технічні умови.

Вхідний файл MAX.DAT містить рядок символів, довжиною не більшою 255 символів.

Записати у вихідний файл MAX.SOL найбільше число з вхідного рядка.

Задача 19. «Новий Рік в епоху інформатизації»

(Заочна олімпіада, 2002-2003)

У одній з шкіл міста Olymprijsk працює лаборант Олена. Вона мріє, щоб Дід Мороз подарував їй у новорічну ніч великого пухнастого білого ведмедика. Зайшовши на офіційний сайт Діда Мороза у Інтернеті www.moroz.did, Оленка вирішила замовити собі новорічний подарунок через on-line систему бронювання подарунків "Yalynka". Але замість вітання "Happy New Year!" вона помітила незрозумілий рядок літер "руHNa pYr eaew!". Це результат роботи жахливого вірусу "AntiFrost", що був написаний хакером Alex'ом. Попри свою шкідливість, вірус має нескладну логіку, і лише переставляє літери у тексті таким чином:

- 1) Вірус окремо опрацьовує голосні та приголосні літери.
- 2) Вірус міняє місцями 1-шу голосну з 2-ою, 3-ю з 4-ою, 5-ту з 6-ою, і т.д. за таким же принципом він міняє місцями і приголосні літери.
- 3) Вірус змінює лише латинські літери, всі інші символи він залишає незмінними.

Голосними літерами вважати (a,o,i,u,y,e)

Оленка зберігла спотворений вірусом текст у файл NOVYJRIK.HTM

Напишіть програму, яка запише початковий (вірний) текст у файл

OLENA.HTM $0 \leq$ Розмір файлу NOVYJRIK.HTM ≤ 60000 байт. У файлі

можуть бути присутні рядки, довжина яких перевищуватиме 255 символів.

Приклад файлу NOVYJRIK.HTM:

```
<THLM>
руННа рYr eaew!
<NIFS TOFO=7>3 Новим Роком!</ZETN>
</THLM>
```

Відповідний файл OLENA.HTM:

```
<HTML>
Happy New Year!
<FONT SIZE=7>3 Новим Роком!</FONT>
</HTML>
```

Задача 20.

(Заочна олімпіада, 2002-2003)

Маленький програміст Петрик вчиться програмувати на мові "CALC". Ця дивовижна мова програмування дозволяє лише обчислювати арифметичні вирази. Мова CALC має такі особливості:

- 1) Змінні позначаються великими латинськими літерами (A,B,C,...,Z)
- 2) Оператор присвоєння має наступний синтаксис:

Змінна = вираз

- 3) Вираз - арифметичний вираз, що містить у собі дужки, числа, змінні та знаки арифметичних операцій + , - , * , / , де ділення відбувається цілочисельно (тобто 7/3 дорівнює 2)

- 4) Виведення реалізується за допомогою оператора: ? вираз

- 5) На початку роботи програми всі змінні мають значення 0

У файлі PETRO.CLC записана правильна програма на мові CALC. Напишіть програму, яка виведе у файл PETRO.OUT результат роботи програми PETRO.CLC.

Гарантується, що значення усіх змінних, виразів, а також будь-якого підвиразу лежатимуть в межах від -10000 до 10000.

Символи пропуску можуть зустрічатися у довільному місті рядка.

Довжина рядків у програмі PETRO.CLC не буде перевищувати 100 символів.

Програма міститиме не більше 50 рядків.

Приклад PETRO.CLC

T = 3

Z = 2

? A

? T

K = T*Z + A - 7

? K

? 3*(5+T) /5

? (((((7))))))

Z=3

Z=-7+1*(-5)+Z

?Z

Результат PETRO.OUT

0

3

-1

4

7

-9

Задача 21. «Розархіватор»

(Заочна олімпіада, перша ліга, 2005-2006)

Юний програміст створив програму-архіватор, яка працює за таким принципом: знаходить у тексті послідовність символів, яка записана кілька разів підряд і замінює цей "ланцюжок" числом повторень та знайденою послідовністю, взятою в дужки. Але програму для розархівації він загубив. Допоможіть юному програмісту скласти програму, яка буде розархівовувати текст. Довжина розархівованого тексту обмежується максимальною довжиною рядка. Текст записано великими латинськими літерами.

Вихідний текст знаходиться у файлі **Z2.DAT**.

Розархівований текст запишіть у файл **Z2.SOL**.

Приклад вхідних і вихідних файлів:

Приклад 1.

Вхідний файл Z2.DAT: KF4(T)G10(N)L

Вихідний файл Z2.SOL: KFTTTTGNNNNNNNNNNL.

Приклад 2.

Вхідний файл Z2.DAT: E3(TY)YU2(NTU)L

Вихідний файл Z2.SOL: EYTYTYTYUNTUNTUL.

Час на тест: 2с

Задача 22.

(Заочна олімпіада, перша ліга, 2005-2006)

Брокеру Тарабарської товарної біржі протягом дня надходять замовлення від клієнтів з повідомленням про те, які акції, в якій кількості та за якою ціною вони бажають купити. Брокер намагається виконати найбільш крупні замовлення, щоб отримати найбільші прибутки. Допоможіть брокеру з'ясувати вартість найкрупнішого замовлення.

Вхідні дані: У першому рядку текстового файлу **broker.dat** записане одне ціле число N – кількість замовлень, отриманих брокером ($0 < N < 100$). Наступні N рядків містять тексти замовлень. Кожне замовлення складається

із слів тарабарською мовою – звертань, назв підприємств, характеристик акцій тощо та цілих чисел, що виражають кількість акцій та їх ціну. Слова записані літерами англійського алфавіту та відділені одне від одного та від чисел пропусками. В замовленні від одного клієнта можуть бути названі декілька видів акцій. Відомо, що для кожного виду акцій обов’язково вказані послідовно два цілих числа – кількість акцій та ціна однієї акції. Довжина кожного замовлення – не більша за 255 символів.

Вихідні дані: У єдиному рядку файлу **broker.sol** записане єдине число – вартість найкрупнішого замовлення.

Приклад 1:

Broker.dat	Broker.sol
3 To me 100 HP for 25 and 50 Audi for 30 and 250 KGB for 20 I want 500 actions BMW for 45 and 300 actions IBM for 35 Ford 200 for 70 euro	33000

Приклад 2:

Broker.dat	Broker.sol
4 gfhgf 10 hgjhkj 5 hjgjhghg 24 nmbmn 50 jhkjhk jhgkjh 56 hgjhgh 2 vnbvnb 789 hgjhg 25 123 hgjhgh hjhjhk 34 hgjg bvcbv 345 hjghg 654	225630

Задача 23. «Текст 1»

(Заочна олімпіада, перша ліга, 2005-2006)

Задано текст довжиною < 1000 символів. Вивести в файл **Z1.SOL** всі слова, що містяться в тексті, в алфавітному порядку.

В тексті можуть бути тільки англійські та українські слова, відокремлені пропуском. У вихідний файл спочатку виводяться англійські, потім українські слова.

Вхідні дані: Текст знаходиться в файлі **Z1.DAT**.

Вихідні дані: Відсортовані слова (кожне в окремому рядку) в файлі **Z1.SOL**.

Приклад вхідних даних:

i take part in an olympiad переклад здійснено програмою pragma

Приклад вихідних даних:

an

i

in

olympiad

part

pragma

take

здійснено

переклад

програмою

Задача 24. «Текст 2»

(Заочна олімпіада, перша ліга, 2005-2006)

Задано текст довжиною < 255 символів. Визначити чи можна з букв, що є в тексті, побудувати задане слово. Кожну букву з тексту можна використовувати тільки один раз.

Вхідні дані: у файлі **Z2.dat** - в першому рядку задано вхідний текст, в другому - задане слово.

Вихідні дані: у стандартний вихідний потік вивести одне слово - "ТАК" чи "НІ".

Приклад вхідних даних: текстовий файл **Z2.dat**:

будівельник

куб

Приклад вихідних даних: ТАК

Задача 25. «Текст 3»

(Заочна олімпіада, перша ліга, 2005-2006)

Задано закодований текст довжиною < 255 символів. Визначити систему кодування та розкодувати текст.

Вхідні дані: у файлі **Z3.dat** - в першому рядку задано вхідний текст.

Вихідні дані: у стандартний вихідний потік вивести розкодований текст. Букви Г і (Г) позначаються однаково.

Приклад вхідних даних: текстовий файл Z3.dat:

бьялюил

Приклад вихідних даних:

команда

Задача 26. «Анаграми»

(Заочна олімпіада, перша ліга, 2006-2007)

У вхідному файлі **Z1.dat** міститься рядок довжиною не більше 255 символів, у якому через один або декілька пропусків слідуєть слова. Знайти всі групи анаграм (*Анаграма* - це слово, яке утворюється з іншого слова перестановкою його букв) у цьому рядку та вивести у файл **Z1.sol** кожну групу з нового рядка. Всі слова повинні йти через пропуск у порядку, у якому вони зустрічаються в рядку.

Однакові слова виводити не можна.

Приклад вхідних і вихідних даних:

Z1.dat

123 321 1234 12345 123456 231 132 3241 123457 123

Z1.sol

123 321 231 132

1234 3241

Задача 27. «Порядок в усьому»

(Заочна олімпіада, перша ліга, 2006-2007)

Розширивши бізнес та переїхавши у новий офіс, фермер Вампіров взявся за наведення елементарного порядку у своїх записах. Він робив їх будь-як і будь-де, тому один запис міг бути записаний на одній сторінці, а наступний – на іншій. Окрім маси, він ще ставив особливі мітки, які записував латинськими буквами. Напишіть програму, яка знищує зайві пропуски між записами, залишаючи на їх місці по одному.

Вхідні дані: у текстовому файлі **order.in** через пропуски записані маси кроликів та, можливо, мітки, які є нероздільними з номером.

Вихідні дані: у текстовий файл **order.out** записати відформатований текст.

Приклад:

Вхідний файл: **order.in**

1 6s 2 3A 4 6

Вихідний файл: **order.out**

1 6s 2 3A 4 6

Задача 28. «Прибуло-вибуло»

(Заочна олімпіада, перша ліга, 2006-2007)

Для своїх бізнесових потреб фермер Дієтенко вів ще один цікавий облік для контролю за кількістю наявних кроликів. Записи його виглядали таким чином:

Add 12

Add 3

Sub 8

.....

„Add 12” свідчило про те, що кількість кроликів на фермі збільшилася на 12, а „Sub 8”, що кількість зменшилася на 8. Складіть програму, яка буде за даними записами визначати кількість кроликів на фермі, якщо відома їх кількість на початку ведення такого обліку.

Вхідні дані: у першому рядку стандартного вхідного потоку міститься ціле N ($N \leq 10000$) – початкова кількість кроликів. Далі йдуть рядки із записами фермера. Натуральні числа, що містяться в цих рядках не перевищують 100, а кількість таких рядків не більша 10000. Гарантується наявність хоча б одного запису.

Вихідні дані: у стандартний вихідний потік вивести наявну кількість кроликів на фермі або -1, якщо у записах міститься помилка, яка на проміжних етапах призводить до від’ємної кількості кроликів.

Приклад вхідних даних.

12
Add 12
Add 3
Sub 8

Приклад вихідних даних.

19

Задача 29. «Двійкові паліндроми»

(Заочна олімпіада, 2008-2009)

З клавіатури вводиться число n (в десятковій системі числення). Вивести з інтервалу $[n, 2*n]$ всі числа (в двійковій системі числення), двійковий запис яких являється паліндромом.

Вхідні дані: з клавіатури вводиться єдине число n ($0 < n < 50000$).

Вихідні дані: вивести на екран двійкові записи чисел-паліндромів із заданого інтервалу, кожне число в новому рядку.

Приклад вхідних і вихідних даних:

Стандартний вхідний потік:

3

Стандартний вихідний потік:

11

101

Пояснення: при $n=3$ інтервал $[3; 6]$ містить 4 числа (3; 4; 5; 6), двійкові записи яких відповідно (11; 100; 101; 110).

Задача 30. «Паліндроми – I»

(Заочна олімпіада, 2008-2009)

Задано рядок, який складається з малих латинських літер. Дозволяється видаляти з рядка певні літери. Скількома різними способами можна при цьому отримати паліндром?

Вхідні дані: заданий рядок знаходиться в файлі `palindrome1.dat`, довжина його не перевищує 30 символів

Вихідні дані: в перший рядок файлу `palindrome1.sol` треба вивести шукану кількість способів отримання паліндрому

Приклад вхідних і вихідних даних:

`palindrome1.dat`

`aab`

`palindrome1.sol`

`4`

Пояснення: паліндром можна отримати, видаливши символи: 1) 1 і 2; 2) 1 і 3; 3) 2 і 3; 4) 3

Задача 31. «Паліндроми – II»

(Заочна олімпіада, 2008-2009)

Задано рядок, який складається з малих латинських літер. Потрібно розбити його на мінімальну можливу кількість паліндромів.

Вхідні дані: заданий рядок знаходиться в файлі `palindrome2.dat`, довжина не перевищує 2000 символів

Вихідні дані: в перший рядок файлу `palindrome2.sol` треба вивести мінімальну кількість паліндромів, на які можна розбити рядок

Приклад вхідних і вихідних даних:

`palindrome2.dat`

`abbacbb`

`palindrome2.sol`

`3`

Пояснення: `abbacbb=abba+c+bb`

Задача 32. «Видалення коментарів»

(Заочна олімпіада, 2008-2009)

Дано текст деякої Паскаль-програми. Ваша задача полягає у тому, щоб видалити з неї всі коментарі та порожні рядки. Як відомо, коментар у паскаль-програмі – це текст, який записаний у фігурних дужках – {}, або у круглих дужках з зірочкою – (* *). Довжина рядка програми не може перевищувати 128 символів. Коментар може займати як частину рядка, так і декілька рядків програми. Вкладені коментарі в тексті відсутні.

Вхідні дані:

У файлі `Z2.dat` міститься текст програми з коментарями.

Вихідні дані:

У файл `Z2.sol` записати текст програми без коментарів. Порожні рядки у файл не записувати.

Приклад:

`Z2.dat:`

`var`

`i:integer; {index }`

`a:array[1..10] of real; (*massiv*)`

`begin {rozvjazok`

`zadachi} for i:=1 to 10 do a[i]:=i;`

`{vyvid massiva} for i:=1 to 10 do`

```

writeln(a[i]);
(*vse*)
end.

Z2.sol:
var
i:integer;
a:array[1..10] of real;
begin
for i:=1 to 10 do a[i]:=i;
for i:=1 to 10 do
writeln(a[i]);
end.

```

Задача 33. «Стек-анаграма»

(Заочна олімпіада, 2008-2009)

Уявімо собі автомат для перетворення слів. На вхід автомата надходить слово. Автомат керується за допомогою двох команд: Push та Pop. Команда Push бере наступну літеру вхідного слова і розміщує її в стеку. Команда Pop бере літеру з вершини стека і дописує її до результату. В результаті на виході автомата утворюється нове слово, яке можливо відрізняється від вхідного порядком літер, в залежності від застосованої послідовності команд Push та Pop. Наприклад:

Вхідне слово: егіс. Позначимо команду Push літерою і, а команду Pop літерою о. В результаті виконання послідовності команд і і о і о і о о на виході утвориться слово гісе.

Очевидно для того, щоб обробити слово з n літер необхідно виконати n команд Push та n команд Pop. В результаті на виході утвориться нове слово з n літер.

Назвемо пару слів стек-анаграмою, якщо друге слово можна отримати шляхом перетворення першого на вищеописаному автоматі.

Завдання: написати програму, що визначає, чи є задана пара слів стек-анаграмою. Якщо є, програма повинна знайти будь-яку послідовність команд, що перетворюють перше слово на друге.

Вхідні дані:

В першому рядку вхідного потоку міститься перше слово. Слово складається лише з маленьких латинських літер. Довжина слова не перевищує 10 літер (та не менше однієї літери). В другому рядку вхідного потоку міститься друге слово. Друге слово теж складається лише з маленьких латинських літер. Довжина другого слова дорівнює довжині першого.

Вихідні дані:

Якщо дані слова є стек-анаграмою, в перший рядок вихідного потоку вивести послідовність команд, що перетворюють перше слово на друге. Команду Push позначити латинською літерою і, команду Pop позначити латинською літерою о. Команди розділити одним або більше пропусками.

Якщо дані слова не є стек-анаграмою, в перший рядок вихідного потоку вивести одну латинську літеру х.

Приклад вхідних даних 1:

mada

adam

Приклад вихідних даних 1:

i i i i o o o o

Приклад вхідних даних 2:

long

nice

Приклад вихідних даних 2:

x

Задача 34. «Decoder»

(Заочна олімпіада, 2009-2010)

Вовочка з Васильком жили в різних будинках і під час осінніх канікул, що дорослі їх назвали Карантином, придумували собі різні забави. Пропоную вашій увазі одну з них. Вовочка відправляє Васильку SMS із словом із N символів. Василько здійснює над словом X ($0 \leq X < N$) циклічних зсувів (циклічний зсув – це коли останній символ переходить на місце першого) і відправляє слово назад до Вовочки. Вовочка повинен визначити скільки циклічних зсувів зробив Василько або чи зробив він помилку при цих зсувах. При невеликих словах ця гра була досить простою, але із збільшенням кількості символів її складність зростає.

Попробуйте скласти програму, яка зможе по двох словах визначити кількість циклічних зсувів, які треба зробити для того, щоб з першого слова отримати друге або встановити, що це зробити неможливо.

Формат вхідних даних. У першому рядку вхідного файлу `decoder.in` міститься натуральне число N ($1 \leq N \leq 250000$) – кількість символів у слові. Другий рядок містить перше слово. Третій рядок містить слово, що було отримане в результаті циклічних зсувів. Слова складаються із символів таблиці ASCII з кодами від 33 до 255.

Формат вихідних даних. У вихідний файл `decoder.out` вивести найменшу кількість циклічних зсувів, що їх треба зробити щоб із першого слова отримати друге або «-1» без лапок у випадку, коли це зробити неможливо.

Приклад вхідних та вихідних даних.

decoder.in

8

karantyn

tynkaran

decoder.out

3

Задача 35. «Степан і паліндроми»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Ім'я вхідного файлу: a.in

Ім'я вихідного файлу: a.out

Степан з дитинства не любив паліндроми. І тому на всіх олімпіадах принципово не розв'язував задачі в яких мова йшла про паліндроми. Вчитель інформатики придумав вихід із складної ситуації. Він запропонував задачу, в якій слід знайти не паліндром. Задано рядок *s*. Знайдіть його найбільший за довжиною підрядок, який не є паліндромом. Нагадаємо (не для Степана), що паліндромом називають рядок, який читається з обох сторін однаково.

Вхідні дані:

Вхідний файл містить рядок *s*. Він містить тільки маленькі букви латинського алфавіту, не пустий, і його довжина не перевищує 100000 символів.

Вихідні дані:

У вихідний файл виведіть відповідь на задачу. Якщо всі підрядки являються паліндромами, то виведіть «NO SOLUTION».

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
abba	abb

Послідовності

Задача 1.

(I обласна олімпіада, 1987 р.)

Написати алгоритм знаходження кількості всіх цілих чисел, більших 10 і менших 1000, десятковий запис яких є неспадною послідовністю цифр.

Задача 2.

(I обласна олімпіада, 1987 р.)

Написати програму, яка підраховує максимальну довжину серії нещасливих білетів. Щасливим білетом вважати той білет, у якого сума правих цифр рівна сумі лівих трьох цифр. Нещасливі білети утворюють серію, якщо вони йдуть один за одним.

Задача 3.

(II обласна олімпіада, 1988 рік)

Скласти програму обчислення суми послідовності $1/1!+2/2!+3/3!+\dots+n/n!$. Обчислити за цією програмою значення сум для $n=9$, $n=37$, $n=68$.

Задача 4.

(IV обласна олімпіада, 1990 рік)

Кіт Матроскін і пес Шарик поїхали на рибалку і вирішили наловити не менше N рибок. За перший день вони виловили по одній риби. Але вночі кіт зголоднів і половину з'їв. У наступні дні Шарик вдавалося виловлювати вдвічі більше, ніж у попередній день і ще дві, а Матроскін виловлював на 5 рибок більше, але вночі кіт з'їдав стільки риби, скільки за всі попередні ночі і ще одну. Допоможіть дядьку Федору скласти алгоритм, за допомогою якого він зміг би дізнатися, коли чекати своїх друзів додому з рибалки.

Задача 5. Неперервний дріб

(V обласна олімпіада, 1991 рік)

Послідовність виду:

$$S_0 = b_0, \quad S_1 = b_0 + \frac{a_1}{b_1}, \quad S_2 = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2}}, \quad S_3 = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3}}} \dots$$

називається послідовністю, що визначає даний неперервний дріб. Складіть алгоритм обчислення S_n для даного n .

Задача 6. «Наближене обчислення»

(VI обласна олімпіада, 1992 рік)

Написати алгоритм знаходження приблизного значення функції $y =$

$\sin(x)$ при малих значеннях за формулою $y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$, використовуючи при цьому мінімальну кількість дій ($n! = 1 * 2 * 3 * \dots * n$).

Задача 7. «Наступність»

(VI обласна олімпіада, 1992 рік)

В таблиці натуральних чисел $A[1], A[2], A[3], \dots, A[n]$ кожне наступне число $A[k+1]$ найменше з чисел, більших $A[k]$ таке, що кожна цифра $A[k]$ є цифрою числа $A[k+1]$. Знаючи $A[1]$, знайти $A[n]$.

Наприклад: 27, 72, 207 і т.д.

Задача 8. «АБРАКАДАБРА»

(XIII обласна олімпіада, 1999 рік)

Послідовність із латинських букв будується так. Спочатку вона пуста. На кожному наступному кроці послідовність подвоюється, після чого до неї зліва дописується чергова буква латинського алфавіту (a, b, c,...). Нище наведені перші кроки побудови послідовності:

Спочатку. Пуста послідовність.

Крок 1. a.

Крок 2. баа.

Крок 3. сбаабаа.

Крок 4. dcbaabaасбаабаа.

.....

Задача полягає в тому, щоб по заданому числу N визначити символ, який стоїть на N -му місці в послідовності, яку отримали після 26-го кроку.

Задача 9. «Дробова частина»

(XIV обласна олімпіада, 2000 рік)

Напишіть програму SHOTPART.*, яка знаходить по заданому n ($1 < n < 1000$) сто десяткових цифр дробової частини числа S_n , де $S_n = 1/2! + 2/3! + 3/4! + \dots + (n-1)/n!$

Вхідні дані

У текстовому файлі SHOTPART.DAT записане єдине число n ($1 < n < 1000$)

Вихідні дані

У текстовому файлі SHOTPART.SOL потрібно записати без пропусків сто цифр дробової частини числа S_n .

Задача 10. «Різні результати»

(XVIII обласна олімпіада, 2004 рік)

Обчислити $1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{1}{2003} - \frac{1}{2004}$ вираз чотирма способами:

послідовно зліва направо;

послідовно зліва направо обчислюється $1 + \frac{1}{3} + \dots + \frac{1}{2003}$ і $\frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2004}$,

потім друге значення віднімають від першого;

послідовно справа наліво;

послідовно справа наліво обчислюють суми, що описані в b), потім проводять віднімання.

У файл `ixxxn4.txt`, де `xxx`-реєстраційний номер учасника, виведіть в окремих рядках результати отриманих обчислень. Починаючи з п'ятого рядка напишіть пояснення отриманих результатів. (Пояснення можна писати в редакторі Блокнот).

Задача 11. «Полосатики»

(XXII обласна олімпіада, 2008 рік)

Наші учені винайшли нову дуже корисну життєву форму, названу *полосатики*. Полосатики – прозорі аморфні амебовидні створіння, які живуть в колоніях. Більшість часу полосатики рухаються. Коли два з них стикаються, замість них виходить новий полосатик. Довгі спостереження, що проводяться нашими ученими, показали, що вага нового полосатика не дорівнює сумі ваг двох зниклих полосатиків, які зіткнулися; проте, вони скоро виявили, що коли два полосатики з масами m_1 і m_2 стикаються, вага полосатика, що вийшов, рівна $2\sqrt{m_1 m_2}$.

Наші учені дуже хочуть знати, до яких меж може впасти сумарна вага даної колонії полосатиків.

Вам потрібно написати програму, що допомагає відповісти на це питання. Ви можете вважати, що 3 або більше полосатиків ніколи не стикаються одночасно.

Вхідні дані

Перший рядок вхідного файлу містить одне ціле число N ($1 \leq N \leq 100$) — кількість полосатиків в колонії. Кожен з наступних N рядків містить одне ціле число від 1 до 10000 — вага відповідного полосатика.

Вихідні дані

Вихідний файл повинен містити один рядок з мінімальною можливою сумарною вагою колонії з точністю до двох знаків після коми (отриманих шляхом округлення).

Задача 12.

(XXIII обласна олімпіада, 2009 рік)

В числовій послідовності a_1, a_2, a_3, \dots відомий перший член, а всі решту знаходяться за формулою $a_i = (a_{i-1})^2 \bmod 10\,000$. Знайдіть N -й член цієї послідовності.

Задача 13. «Послідовність Хеммінга»

(Заочна олімпіада, 2000-2001)

Послідовністю Хеммінга називається послідовність всіх натуральних чисел великих одиниці, які мають інших простих дільників крім 2, 3 і 5, в якій кожне наступне число більше попереднього. Ось її початок:

2 3 4 5 6 8 9 10 12 15 16 18 ...

Написати програму, що виводить у вихідний потік перший n чисел послідовності, розділені перекладом рядка.

Вихідні дані: у першому рядку вхідного потоку міститься число n . $1 \leq n \leq 1500$.

Задача 14. Послідовність

(Заочна олімпіада, 2001-2002)

Дано натуральне число k (≤ 100000). Надрукувати k -ту за порядком цифру послідовності 149162536..., в якій виписані підряд квадрати натуральних чисел.

Вхідні дані: у файлі **INPUT.DAT** натуральне число k .

Вихідні дані: вивести у файл **OUTPUT.DAT** k -ту цифру в послідовності.

<i>Приклад 1.</i>	<i>Приклад 2.</i>
INPUT.DAT	INPUT.DAT
10	9000
OUTPUT.DAT	OUTPUT.DAT
4	3

Задача 15. Рядок

(Заочна олімпіада, 2001-2002)

Дано рядок із N символів ($1 \leq N \leq 1000$). Далі виписуються спочатку всі однолітерні підрядки даного рядка, потім дволітерні і т.д. Наприклад, якщо рядок `abcd`, то виписані підрядки:

`a,b,c,a,d,ab,bc,ca,ad,abc,bca,cad,abca,bcad,abcd`

або забравши коми:

`abcdabbccaadabcbcsacadabcabcdabcd`

Необхідно вказати, яка буде K -та літера у виписаному рядку.

Вхідні дані: число K та вхідний рядок.

Вихідні дані: шуканий символ

Приклад:

7

`abcd`

Відповідь:

`b`

Задача 16. «Сегменти»

(Заочна олімпіада, 2003-2004)

Визначимо множину S наступним чином. $S[0]$ - відрізок $[0;1]$. Розділимо відрізок $[0;1]$ на три частини точками $1/3$ і $2/3$ і видалимо проміжок $(1/3;2/3)$. Отримаємо $S[1]$, що містить два відрізки $[0;1/3]$ та $[2/3;1]$. Кожний з них розділимо на три частини (точками $1/9$ і $2/9$ перший та $7/9$ і $8/9$ другий) та видалимо середні частинки, тобто відрізки $(1/9;2/9)$ та $(7/9;8/9)$. Ті відрізки, що залишаться утворюють $S[2]$. І т.д.

За відомим числом m із запропонованого набору дробів виду a/b визначити ті, які належать вказаній множині $S[m]$.

Технічні умови:

Вхідні дані знаходяться в файлі "PIESEC.DAT", який має структуру:

в першому рядку єдине число m ($0 \leq m \leq 100$);
далі записані дроби у вигляді a/b ($0 < a, b \leq 6000000$) по одному в
кожному рядку.

Результат - знайдені дроби вивести у файл "PIESEC.SOL" по одному в
кожному рядку.

Приклад 1:

PIESEC.DAT	PIESEC.SOL
1	1/1
1/1	1/3
1/2	
1/3	

Приклад 2:

PIESEC.DAT	PIESEC.SOL
3	1/3
1/2	1/4
1/3	1/9
1/4	
1/5	
1/6	
1/7	
1/8	
1/9	

Приклад 3:

PIESEC.DAT	PIESEC.SOL
25	172907/2000907
162001/1990001	4171979/5999979
172907/2000907	
4171979/5999979	

Задача 17. «Числова послідовність»

(Заочна олімпіада, 2003-2004)

Дана послідовність натуральних чисел 7, 11, 13, 14, 19, 21, 22, 25,

Написати програму, яка по заданому N ($1 \leq N \leq 2000000000$) знаходить N -й член цієї послідовності.

Вхідні дані: число N .

Вихідні дані: відповідне число.

Приклад вхідних даних:

7

22

Приклад вихідних даних:

Задача 18. «Послідовність Хеммінга»

(Заочна олімпіада, 2003-2004)

Числами Хеммінга називаються натуральні числа, які мають тільки дільники 2, 3 і 5 в будь-якому цілому невід'ємному степені. Перші члени цієї послідовності виглядають таким чином: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15 ...

Першому числу цього ряду відповідають нулеві степеня даних дільників. Скласти програму, яка по заданому N ($N \leq 1600$) видає N -й член послідовності.

Обмеження: розмір файлу менший 3 КБ.

Вхідні дані: число N .

Вихідні дані: відповідне число.

Приклад вхідних даних:

10

Приклад вихідних даних:

12

Задача 19. «Стрічка Фібоначі»

(Заочна олімпіада, 2003-2004)

Визначимо ряд стрічок Фібоначі наступним чином.

$S[1] = "a"$, $S[2] = "b"$, $S[i] = S[i-2] + S[i-1]$,

де операція "+" означає "склеювання" двох стрічок в одну.

Необхідно визначити, яка літера буде n -ою літерою стрічки $S[45]$.
 $1 \leq n \leq 1134903170$.

Вхідні дані:

Вхідний потік містить в першій стрічці єдине число - значення n .

Вихідні дані:

У вихідний потік слід помістити одну літеру, що знаходиться в позиції n в стрічці $S[45]$.

Приклад:

Вхідний потік: 701408733

Вихідний потік: b

Задача 20. «Цікава послідовність»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Василько познайомився із двійковою системою числення і був надзвичайно вражений, що у ній є тільки дві цифри: 0 і 1. Але добре знаючи десяткову систему числення він помітив, що степені числа 10 також складаються лише із нулів та одиниць. Ось одна із його послідовностей, яка потребує неабиякої майстерності, щоб відповісти на Василькові питання. Побудуємо послідовність, яка буде складатися із степенів 10: 110100100010000... . Напишіть програму, яка зможе за вказаним порядковим номером визначити цифру цієї послідовності.

Вхідні дані:

Перший рядок стандартного вхідного потоку містить N ($1 \leq N \leq 65535$) - кількість тестів. В i -му із N наступних рядків записане ціле K_i - номер позиції в послідовності ($1 \leq K_i \leq 2^{31}-1$).

Вихідні дані:

У стандартний вихідний потік вивести через пропуск послідовність цифр, які відповідають позиціям K_i .

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
4 3 14 7 6	0 0 1 0

Рекурсія, перебір

Задача 1.

(II обласна олімпіада, 1988 рік)

Селище складається з N будинків, розташованих уздовж прямої дороги з одного сторони на рівних відстанях. У селищі проводять телефонний зв'язок. У таблиці T вказано, скільки телефонних апаратів треба встановити в кожному будинку. Написати алгоритм, який визначає, в якому будинку потрібно встановити АТС, щоб сумарна відстань від АТС до всіх телефонних апаратів була мінімальною. Кожен телефон пов'язаний з АТС окремим проводом.

Задача 2.

(II обласна олімпіада, 1988 рік)

Скласти алгоритм, що вибирає всі чотиризначні числа, у запису яких немає цифр 1 і 9.

Задача 3.

(II обласна олімпіада, 1988 рік)

В некотором городе все улицы проходят с севера на юг и с запада на восток. На некоторых перекрестках установлены телефоны-автоматы. Для каждого такого перекрестка заданы его координаты $X[i]$, $Y[i]$ и количество автоматов на нем $T[i]$, $i=1, \dots, N$. Написать алгоритм, определяющий координаты АТС, при которых сумма длин проводов, идущих от АТС к каждому автомату минимальна. Провода разрешается прокладывать только вдоль улиц (то есть, с севера на юг или с запада на восток).

Задача 4.

(III обласна олімпіада, 1989 рік)

Скласти алгоритм, який одержує на вході число K і видає K -е по порядку чотиризначне число, у якого ніякі дві цифри не рівні між собою. Першим таким числом вважати 0123.

Задача 5.

(III обласна олімпіада, 1989 рік)

Скласти рекурсивний алгоритм обчислення найбільшого спільного дільника двох натуральних чисел.

Задача 6.

(III обласна олімпіада, 1989 рік)

На шаховій дошці M / N розставити найменшу кількість ферзів так, щоб всі клітини дошки знаходилися під боєм хоча б одного ферзя.

Задача 7.

(III обласна олімпіада, 1989 рік)

На площині дано N різних точок. Серед них вибрати три такі, щоб трикутник, вершинами якого вони служать, мав найбільшу площу.

Задача 8.

(III обласна олімпіада, 1989 рік)

Написати програму виведення всіх тризначних десяткових чисел, сума цифр яких дорівнює даному цілому числу M .

Задача 9. Нумерація клітин

(VI обласна олімпіада, 1992 рік)

Одна з клітин прямокутної дошки $M \times N$ зафарбована. Пронумеруйте інші клітини числами 1, 2, 3, ... так, щоб дві клітини з сусідніми номерами мали спільну сторону.

Задача 10. «Три зірочки в шестизначному числі»

(VII обласна олімпіада, 1993 рік)

Замінити в числі 523*** зірочки такими трьома цифрами, щоб одержане шестизначне число ділилося одночасно на 7, 8, 9.

Задача 11. «Оплата покупки»

(VII обласна олімпіада, 1993 рік)

Купони багаторазового користування України мають вартість $A_1, A_2, A_3, \dots, A_n$ карбованців. Як оплатити покупку вартістю в S карбованців, використовуючи найменшу кількість купонів.

Задача 12. «Сума кубів»

(VII обласна олімпіада, 1993 рік)

Скільки натуральних чисел, що не перевищують n , можна подати у вигляді суми кубів двох натуральних чисел? Якщо можна, то показати як. Враховувати тільки одне можливе представлення.

Задача 13.

(X обласна олімпіада, 1996 рік)

Магічним квадратом зветься такий числовий квадрат, заповнений різними натуральними числами, у якого всі суми чисел, розташованих на довільній горизонталі, вертикалі або діагоналі, рівні між собою. Приклад магічного квадрата:

492

357 (4+9+2=3+5+7=8+1+6=4+3+8=9+5+1=2+7+6=4+5+6=2+5+8=15)

816

Знайти всі магічні квадрати 3×3 , що складаються з натуральних чисел та задовольняють трьом вимогам:

1) якщо до кожного числа додати 1, то всі числа стануть простими, і квадрат залишиться магічним.

2) якщо від кожного числа відняти 1, то всі числа стануть простими, і квадрат залишиться магічним.

3) кожне число магічного квадрата не перевершує 1996.

Зверніть увагу: наведений приклад магічного квадрата цим вимогам не задовольняє! Квадрати, які можна отримати один з одного симетріями, вважайте різними.

Вивести на екран і в файл SQUARE.OUT кількість знайдених квадратів, а також самі квадрати в зручному для перевірки вигляді.

Задача 14. «Шахівниця»

(XI обласна олімпіада, 1997 рік)

В квадратній дошці розміром $n \times n$ клітинок вирізали деякі дві клітинки. Отже, дістали дошку, що складається з $n^2 - 2$ клітинок. Маємо також $(n^2 - 2)/2$ дощечки розміром 1×2 клітинок, кожна з яких дозволяється класти на дошку так, щоб накривались рівно дві її клітинки. Дощечки не можуть перекриватися.

Дано: n – довжину сторони дошки (в клітинках)

$(x_1, y_1), (x_2, y_2)$ – координати вирізаних клітинок,

$1 \leq x_1, y_1, x_2, y_2 \leq n$.

Отримати: відповідь на питання – чи можна накрити дощечками розміром 1×2 всі клітинки заданої дошки? Якщо можна, то вказати, як це зробити.

Задача 15. «Відрізок»

(XII обласна олімпіада, 1998 рік)

Задане число N - кількість точок на площині, та числа $X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N$ – координати точок. Потрібно скласти програму LINE.* , яка визначає координати кінців відрізка найменшої довжини та довжину відрізка. На екрані зобразити точки і шуканий відрізок.

Задача 16. «Міжнародна конференція»

(XII обласна олімпіада, 1998 рік)

Вас найняли для того, щоб визначити місця дипломатів за столом обговорень міжнародної конференції. На конференцію запрошені по одному дипломату з N різних країн світу. Кожен дипломат знає від однієї до M мов. Дипломати, які не знають спільної мови, не можуть розмовляти один з одним. До того ж, деякі країни проголосили, що не будуть підтримувати дипломатичних стосунків з деякими іншими, тобто представники цих країн не будуть розмовляти один з одним. Ваша задача полягає в розробці програми DIPLOMAT.*, що визначає місця за столом для дипломатів таким чином, щоб кожен міг розмовляти з обома своїми сусідами, які сидять ліворуч та праворуч від нього.

Стіл, що використовується, – круглий і розрахований на N персон. Дипломат може спілкуватись з дипломатом, який сидить ліворуч однією мовою, а з дипломатом що сидить праворуч - іншою.

Задача 17. «Числа (+ -)»

(XVI обласна олімпіада, 2002 рік)

У рядок виписано N чисел ($1 \leq N \leq 40$). Потрібно перед кожним числом поставити знак "+" або "-" так, щоб значення отриманого виразу було рівне K .

Вхідні дані:

у першому рядку числа N і K .

у другому N цілих чисел, у межах $[0..20]$.

Вихідні дані:

результуючий вираз, в якому знаки та числа відокремлені одним пропуском.

Приклад:

3 7

10 14 3

Відповідь:

$$- 10 + 14 + 3 = 7$$

Задача 18. «Трикутники»

(XVII обласна олімпіада, 2003 рік)

У текстовому файлі TRIANGLE.DAT через пропуск записані 10 різних натуральних чисел (кожне з яких не перевищує 5000). Потрібно побудувати алгоритм і реалізувати його у вигляді програми TRIANGLE.PAS (BAS чи CPP), що прочитає дані з файла TRIANGLE.DAT, підрахує кількість різносторонніх трикутників, довжинами сторін яких служать дані числа та результат виведе у файл TRIANGLE.SOL.

Приклад:

TRIANGLE.DAT

1 3 5 9 15 25 45 75 125 500

TRIANGLE.SOL

0

Задача 19. «Симетричні простачки»

(XVII обласна олімпіада, 2003 рік)

Число називається простим, якщо воно має лише два дільники: 1 і саме число. Назвемо число "симетричним", якщо його десятковий запис читається однаково зліва направо і справа наліво (151, 2002, ...). У файл SYMETR.* (де * означає розширення дозволеної мови програмування PAS для Pascal, CPP для C++, BAS для Basic) запишіть програму, яка підрахує кількість простих "симетричних" чисел у вказаному діапазоні $[n, m]$, де $10 < n < m < 100\ 000\ 000$.

Вхідні дані:

В першій стрічці вхідного потоку міститься два числа n та m розділені одним пропуском.

Вихідні дані:

Вивести в першу стрічку вихідного потоку єдине число, що є кількістю простих "симетичних" чисел із вказаного діапазону.

Приклади даних:

Вхідні дані

10 100

20000 345678

10 100000000

Вихідні дані

1

67

777

Задача 20. «Шифровка»

(XVIII обласна олімпіада, 2004 рік)

Відомо, що учасники обласної олімпіади Дунець Павло, Печений Олександр та Райчук Антон на зимові канікули запрошувалися на збори за результатами успішного виступу на минулорічній Всеукраїнській олімпіаді, що відбулася в Донецьку. Для них і всіх інших учасників обласної олімпіади журі склало шифровку, яка записана у файлі text.dat. Спосіб шифрування не відомий, але відомо, що літери та інші символи не змінювалися. Змінено лише місце їх розташування. Вам слід отримати розшифрований текст і записати його у файл ixhxn2.txt, де xxx – Ваш реєстраційний номер (при потребі реєстраційний номер уточніть у чергового члена журі).

Задача 21. «Телепортація»

(XVIII обласна олімпіада, 2004 рік)

Розглянемо прямокутну декартову систему координат O_{xy} (будемо вважати її моделлю Всесвіту). Припустимо, що в точці (x_1, y_1) знаходиться космічний корабель. За одну секунду з точки (x, y) він може телепортуватися в точки $(x+C, y+C)$, $(x+C, y-C)$, $(x-C, y+C)$, $(x-C, y-C)$, де C - довільне

натуральне число. Який мінімальний час знадобиться кораблю для того, щоб досягти точки (x_2, y_2) ?

Введення

Ваша програма повинна вводити з клавіатури числа x_1, y_1, x_2, y_2 . Числа x_1 і y_1 повинні бути введені в першому рядку, числа x_2 і y_2 - в другому рядку.

Виведення

Якщо можна досягти точку (x_2, y_2) з точки (x_1, y_1) , то виведіть на екран мінімальний час в секундах, необхідний для цього. В протилежному випадку виведіть на екран число 0.

Приклад

<i>Введення</i>	<i>Виведення</i>
0	02
0 2	

В даному прикладі до успіху приводить така послідовність телепортацій:

$(0, 0) - (1, 1) - (0, 2)$.

Обмеження.

$0 \leq x_1, y_1, x_2, y_2 \leq 1000000000$; всі числа при введенні цілі; точки (x_1, y_1) і (x_2, y_2) не співпадають.

Зауваження.

Якщо ваша програма видає однакову відповідь на всіх тестових прикладах, то вона отримає 0 балів.

Задача 22. «Дід Мороз і комп'ютерні хулігани»

(XIX обласна олімпіада, 2005 рік)

На поштову скриньку Діда Мороза надходять замовлення на подарунки у вигляді текстового файлу. Поштова програма працює так, що вихідний текст спочатку кодується, а потім пересилається паралельно з оригіналом. На комп'ютері Діда Мороза програма розкодує файл та порівнює з оригіналом. Як завжди, комп'ютерні хулігани, перед Новим роком хочуть заявити про

себе, "запустивши" в мережу новий комп'ютерний вірус. З усієї кількості надісланих листів непошкодженими дійшли лише два листи-оригінали та закодовані листи з замовленнями.

Допоможіть Діду Морозу розкодувати та прочитати решту листів.

Технічні умови:

У файлі SURPRISE.ORI знаходиться оригінал листа, у файлі SURPRISE.KOD закодований текст.

Ваша програма повинна прочитати дані з файлу SURPRISE.KOD і вивести оригінал листа у файл SURPRISE.ORI

Задача 23. «Сіамські простачки»

(XX обласна олімпіада, 2006 рік)

Число називається простим, якщо воно має лише два дільники: 1 і саме число.

Назвемо число "сіамським простачком", якщо:

- воно є простим;
- його десятковий запис читається однаково зліва направо і справа наліво;
- перші $(k \text{ div } 2)+1$ цифр і останні $(k \text{ div } 2)+1$ цифр утворюють прості числа, де k – кількість цифр даного числа.

Наприклад: число 3541453 є "сіамським простачком", бо число 3541453 є простим, а також кожне з чисел 3541 та 1453 є простими числами.

Напишіть програму, яка виводить "сіамські простачки" з вказаного діапазону $[n, m]$, де $100 < n < m \leq 10000000$ або повідомлення **NO**, якщо таких чисел у вказаному діапазоні немає.

Вхідні дані:

В першій рядку текстового файлу **syamsimp.dat** міститься два цілих числа n та m розділені одним пропуском.

Вихідні дані:

Вивести в текстовий файл **syamsimp.sol** “сіамські простачки” з вказаного діапазону в порядку зростання. Кожне нове число в новому рядку або повідомлення **NO**, якщо таких чисел у вказаному діапазоні немає.

syamsimp.dat	syamsimp.sol
101 352	131 313
2007 7002	NO

Задача 23. «І знову дужки»

(XX обласна олімпіада, 2006 рік)

Даний рядок, який складається із символів “()[]{}”. Визначити, яку найменшу кількість символів необхідно вставити у цей рядок, щоб він перетворився на правильний вираз з дужками.

Вхідні дані: у першому рядку вхідного потоку міститься заданий рядок символів. Довжина рядка не перевищує 100 символів.

Вихідні дані: у перший рядок вихідного потоку вивести одне число – найменшу кількість символів, які необхідно вставити в даний рядок символів, щоб він перетворився на правильний вираз з дужками.

Приклад вхідних даних:

{() }

Приклад вихідних даних, що відповідають вхідним:

2

Задача 24. «Прямокутник»

(XXII обласна олімпіада, 2008 рік)

Тарас вирішив розрізати прямокутну сторінку в клітинку розміру $n \times m$ клітинок на квадрати. Спочатку він відрізує як можна більший квадрат, використовуючи один прямий розріз. Потім він прибирає отриманий квадрат і повторює дію з прямокутником, що залишився. Так поступати (весь час

відрізуючи як можна більший квадрат) Тарас продовжує до тих пір, поки прямокутник, що залишився, не буде квадратом.

Ваше завдання – написати програму, яка по введених значеннях n і m ($n < 10000$, $m < 10000$) обчислює, як багато квадратів Тарас отримає вирізуванням з прямокутника описаним вище шляхом.

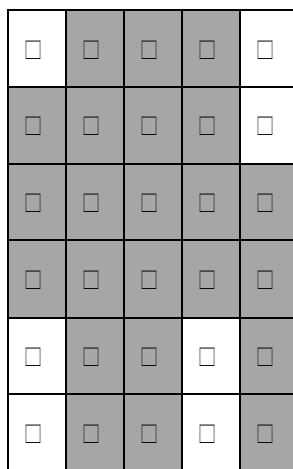
Приклади

Введення	Виведення	Коментар
3 7	5	Прямокутник був розрізаний на квадрати з довжинами 3, 3, 1, 1, 1

Задача 25. «Стрілочки»

(XXII обласна олімпіада, 2008 рік)

На папері намалювали картате (в клітинку) поле $N \times M$ клітинок. У кожній клітинці намалювали стрілку в одному з чотирьох напрямів «управо», «вгору», «вліво» або «вниз».



Далі в деяку клітинку цього поля ставлять фішку. Потім цю фішку зміщують в сусідню клітинку у напрямі стрілки, намальованої в клітинці, де стоїть фішка. Потім її знову зміщують по стрілці, намальованій в тій клітці, де вона опинилася. Так продовжується до тих пір, поки фішка не опиниться за межами поля. Проте можливо, що фішка нескінченно ходитиме по полю і ніколи не вийде за його межі.

Напишіть програму, яка по заданому полю визначить кількість клітинок, почавши з яких фішка ніколи не покине межі поля.

Формат вхідних даних

У вхідному файлі задані спочатку розміри поля – число рядків N і число стовпців M ($1 \leq N \leq 1000$, $1 \leq M \leq 1000$). Далі йде N рядків по M чисел в кожному, які задають напрями стрілок в клітинках. Число 1 позначає стрілку управо, 2 – вгору, 3 – вліво, 4 – вниз. Числа в рядку розділяються пропусками.

Формат вихідних даних

У вихідний файл виведіть одне число – кількість клітинок, почавши з яких фішка ніколи не покине межі поля.

Приклади

f.in	f.out	Зауваження
6 5 3 1 1 4 2 1 2 4 3 1 4 2 1 1 4 1 2 3 3 3 3 1 4 4 4 2 2 3 4 2	23	Цей приклад відповідає приведеному малюнку. Клітинки, почавши з яких, фішка ніколи не покине меж поля на малюнку виділені сірим кольором.
2 2 1 2 3 4	0	

Задача 26.

(XXIII обласна олімпіада, 2009 рік)

Візьмемо будь-яке натуральне число и додамо до нього суму його цифр. Отримане число називається породженим, а перше число – генератором породженого числа.

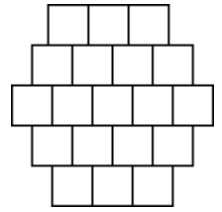
Наприклад, $47 + 4 + 7 = 58$. 58 – породжене число, а 47 – його генератор.

Знайдіть для даного натурального числа N ($N < 10^{16}$) його генератори. У першому рядку вивести кількість, у другому через пропуск генератори у порядку їх зростання.

Задача 27.

(Заочна олімпіада, 2000-2001)

На рисунку зображена фігура. Існує спосіб заповнення клітинок фігури натуральними числами від 1 до 19 так, щоб сума елементів в довільній діагоналі дорівнювала 38. В фігурі 15 правильних діагоналей. Всі числа в клітинках повинні бути різними.



Завдання: в файл `MAGIK.SOL` необхідно вивести всі можливі заповнення фігури відповідно до умови задачі (симетричні заповнення вважаються різними).

Вихідні дані: кожне заповнення фігури у вихідному файлі має займати шість рядків:

перший – 3 числа через пропуски; другий – 4 числа; третій – 5 чисел; четвертий – 4 числа; п'ятий – 3 числа; шостий – пустий рядок. Далі наступне заповнення.

Примітка: кожне правильне заповнення додає бали, а неправильне – віднімає.

Задача 28.

(Заочна олімпіада, 2000-2001)

Мається 10 фішок викладених в ряд. Кожна фішка біла з одного боку, і чорна з іншого. Хід робиться так: вибирається будь-яка фішка, і вона перевертається, разом зі своїми сусідами. Необхідно за найменше число ходів зробити так, щоб всі фішки лежали білою стороною вгору.

Наприклад:

ЧБЧ

БЧЧ

БББ

Ваша програма повинна записати у перший рядок вихідного потоку число N кількість ходів, а у другому рядку N чисел розділених пробілами (номери вибраних фішок на кожному ході)

Вихідні дані: у першому рядку вхідного потоку міститься рядок з 10 символів, яка визначає початковий стан всіх фішок (Б - біла, Ч - чорна)

Задача 29. Камінь

(Заочна олімпіада, 2001-2002)

У місті Хмельницькому на місці першого поселення встановили великий камінь у вигляді куба. На західній стороні каменя встановили табличку з відповідним написом. Та от халепа, закріпити табличку потрібно було на східній стороні. А треба сказати, що закріпили табличку надійно. Отже потрібно перекочувати камінь з боку на бік так, щоб табличка опинилася на східній стороні каменя в правильному положенні (не боком і не догори ногами), а також так, щоб камінь ніколи не ставав табличкою вниз. Під час перекочування камінь не повинен ставати на клумби і після перекочування повинен знаходитися на тому самому місті, що й до. Вам потрібно визначити найкоротшу послідовність перекочувань, яка приведе до поставленої цілі при заданому розміщенні клумб. Цим самим ви добре допоможете робітникам, які будуть виконувати цю роботу.

Поле дії -- 11 на 11 клітин з стороною, що рівна стороні каменя (вид зверху). Кожна клітина може бути зайнятою клумбою або вільною. Верх поля -- північ. Камінь стоїть посередині поля в клітині з координатами (6,6). Табличка з східної сторони в правильному положенні. Перекочування каменя переводить його на відповідну сусідню клітину з відповідною зміною положення таблички.

У вхідному потоці міститься 11 стрічок по 11 непробільних символів у кожній. Символи можуть бути тільки '.' (крапка) та '0' (нуль). Символи можуть бути довільно розділені пропусками. Крапка позначає вільну клітину, нуль позначає клітину, що зайнята клумбою. Перша стрічка -- північна сторона поля. Відповідно зліва захід, справа схід. Шоста позиція в шостій стрічці завжди 'X' (велика латинська ікс), що позначає початкове положення каменя.

Вивести у вихідний потік стрічку символів, що задають послідовність перекочувань каменя таким чином:

N - перекочування на північ;

E - на схід;

S - на південь;

W - на захід.

Літери обов'язково великі латинські. Стрічка не повинна містити ніяких символів, крім даних чотирьох. Вивести пусту стрічку, якщо неможливо досягти поставленої цілі задовольняючи всі задані умови.

Приклад.

Вхід:

.....

.....

.....

... 0 0 . 0 0 ...

..... X

... 0 0 . 0 0 ...

.....

.....

.....

.....

Вихід: NNENWWSESS

Задача 30.

(Заочна олімпіада, 2001-2002)

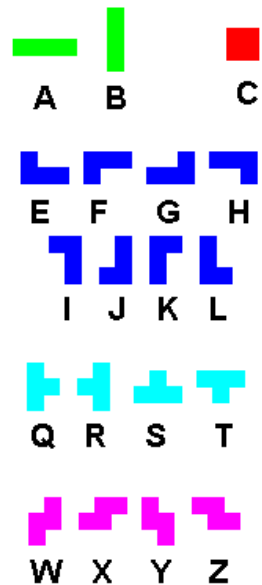
Маємо 19 типів різнокольорових тетрамонічних фігурок (дивіться малюнок).

З клавіатури в першому рядку вводяться два числа N та M – розміри прямокутника, який необхідно заповнити фігурками; в другому рядку 19

чисел – наявна кількість фігурок кожного типу. (N - кількість рядків. $N \leq M < 24$)

В заповненому прямокутнику повинні зустрічатися всі кольори фігурок, і не повинно бути пустих клітинок.

Напишіть програму, яка виводить на екран монітору кількість варіантів заповнення даного прямокутника даною кількістю фігурок. Також виведіть на екран один з варіантів.



Приклад вхідних даних:

3 8

1 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1

Приклад вихідних даних:

1

QZZTTTII

QAAAACCI

QQZZTCCI

Задача 31. «Planeta X»

(Заочна олімпіада, вища ліга, 2005-2006)

Чи єдині ми у Всесвіті? Це питання здавна хвилює людство. Після недавніх сенсаційних розкопок на планеті X13 були знайдені листки з таємничими символами. Довгі дослідження вчених показали, що надписи можуть бути числовими виразами. Якщо їхній висновок вірний, то на планеті X13 багато тисячоліть назад могли існувати розумні істоти, оскільки вони вміли рахувати.

Вчені змогли зрозуміти, що означають написані символи, і перевели все на звичайну мову – мову цифр, дужок, знаків арифметичних дій та рівності. Крім того, стало ясно, що на даній планеті можливі жителі знали три операції: додавання, множення та віднімання (вони ніколи не використовували "унарний мінус": замість "-1" вони записували "0-1"). Також вченим вдалося встановити, що арифметичні операції не мали

пріоритетів, а вирази просто обчислювалися зліва направо, якщо в них не було дужок: наприклад $3+3*5$ у них дорівнювало 30, а не 18.

Нажаль X-планетяни, назвемо їх так, маючи пізнання в математиці, погано розумілися на хімії. Чорнила, якими були списані листки не були стійкими до часу і, мені також показалося це дивним, знаки операцій зникли - залишилися лише замість них пропуски. Якщо теорія вчених вірна, то коли замість пропусків поставити знаки арифметичних операцій - вираз буде правильним. Наприклад, на листку є напис " $18=7 (5 3) 2$ ". Можлива така розстановка знаків: " $18=7+(5-3)* 2$ ". В той же час, якщо попадався запис " $5=3 3$ ", то брати по розуму, напевне, мали на увазі щось інше...

Ви повинні написати програму, що знаходить розстановку знаків або повідомити, що такої не існує.

Вхідні дані:

Перший рядок вхідного файлу **Planeta.in** складається із натурального числа не більшого 2^{30} , знака рівності, і послідовності натуральних чисел (не більше 10), добуток яких не більший також 2^{30} . Деякі групи чисел можуть бути в дужках. Довжина рядка не перевищує 80 символів. Між сусідніми числами завжди є хоча б один пропуск, у всіх інших випадках може бути будь-яка кількість пропусків (в тому числі і 0). Всередині числа X-планетяни пропусків не ставили.

Вихідні дані:

У вихідний файл **Planeta.out** необхідно вивести один рядок, що містить утворену рівність або "-1", якщо розстановка знаків є неможливою для правильної рівності. Вихідний рядок не має містити пропусків.

Наприклад:

Planeta.in	Planeta.out
$18=7 (5 3) 2$	$18=7+(5-3)*2$
$5= 3 3$	-1

Задача 32. «Пошук імен»

(Заочна олімпіада, перша ліга, 2006-2007)

Кожному кролику Дістенко давав унікальне ім'я. Звичайно, час від часу виникала ситуація, коли імена звільнялися і їх можна було знову використати. Отже, наш фермер має список усіх можливих імен і список імен наявних кроликів. Складіть програму для того, щоб зі списку можливих імен вибрати лише ті, що є вільними і їх можна використати знову.

Вхідні дані: у першому рядку стандартного вхідного потоку задається ціле N ($N \leq 10000$) – кількість можливих імен. У наступних N рядках записані самі імена, що містять лише символи латинського алфавіту і мають довжину не більшу 20. Потім в новому рядку міститься ціле K ($K \leq 10000$) – кількість використаних імен. Далі йде перелік цих імен в окремих рядках.

Вихідні дані: в окремих рядках стандартного вихідного потоку вивести ті імена, що є вільними для використання.

Приклад вхідних даних.

```
4
Mario
Nensi
Gig
Joss
2
Joss
Nensi
```

Приклад вихідних даних.

```
Mario
Gig
```

Задача 33. «Судоку»

(Заочна олімпіада, 2008-2009)

Поле для гри в судоку має розмір 9x9 клітинок і умовно розбите на 9 квадратів розміром 3x3.

Кожна клітинка є або пустою, або містить число від 1 до 9.

Приклад:

389|7..|2..

672|.5.|.1.

.14|6.9|.3

23|.5..|871

85|.37|.49

497|81|.3.5

.65|482|13.

1.3|.7.|46.

74.|1.3|59.

Вам необхідно розставити в пусті клітинки числа від 1 до 9 так, щоб

- 1) в кожному рядку числа не повторювались
- 2) в кожному стовпчику числа не повторювались
- 3) в кожному з 9-ти умовних квадратів числа не повторювались

Для вищенаведеного прикладу відповідь наступна:

389|741|256

672|358|914

514|629|783

236|594|871

851|237|649

497|816|325

965|482|137

123|975|468

748|163|592

Вхідні дані:

У стандартному вхідному потоці записані 9 рядків по 9 чисел.(числа розділені пропусками). Пуста клітинка позначена числом 0. Нулів (пустих клітинок) буде не більше ніж 29

Вихідні дані:

у стандартний вихідний потік запишіть відповідь до заданого sudoku - 9 рядків по 9 чисел в кожному (розділені пропусками!!!)

Приклад вхідних даних:

3 8 9 7 0 0 2 0 0

6 7 2 0 5 0 0 1 0

0 1 4 6 0 9 0 0 3

2 3 0 5 0 0 8 7 1

8 5 0 0 3 7 0 4 9

4 9 7 8 1 0 3 0 5

0 6 5 4 8 2 1 3 0

1 0 3 0 7 0 4 6 0

7 4 0 1 0 3 5 9 0

Приклад вихідних даних:

3 8 9 7 4 1 2 5 6

6 7 2 3 5 8 9 1 4

5 1 4 6 2 9 7 8 3

2 3 6 5 9 4 8 7 1

8 5 1 2 3 7 6 4 9

4 9 7 8 1 6 3 2 5

9 6 5 4 8 2 1 3 7

1 2 3 9 7 5 4 6 8

7 4 8 1 6 3 5 9 2

Задача 34. «Запросто»

(Заочна олімпіада, 2008-2009)

Дано n чисел, $3 \leq n \leq 1000$, $-1000 \leq x_i \leq 10000$. Знайти серед них два таких числа x_a та x_b , що їх сума дорівнює сумі всіх інших чисел, що залишилися.

Вхідні дані:

В першому рядку вхідного потоку міститься одне число n . В другому рядку вхідного потоку міститься n чисел x_i , розділених пропуском.

Вихідні дані:

В перший рядок вихідного потоку вивести два числа x_a та x_b розділених одним або більше пропусками. Якщо неможливо знайти таких два числа — вивести числа 0 та 0.

Приклад вхідних даних 1:

```
5
-5 2 5 3 5
```

Приклад вихідних даних 1:

```
2 3
```

Приклад вхідних даних 2:

```
4
2 6 -4 1
```

Приклад вихідних даних 2:

```
0 0
```

Час роботи програми не повинен перевищувати 5 сек на Intel Core 2 Duo E2180.

Задача 35. «Куб»

(Заочна олімпіада, 2010-2011)

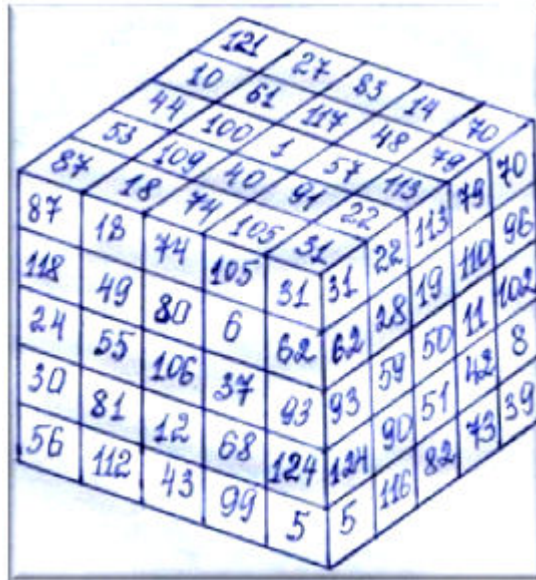
Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Ім'я вхідного файлу: b.in

Ім'я вихідного файлу: b.out

Дано трьохвимірний масив А цілих чисел розмірності N, M, K який зручно представити у вигляді куба, що складається з кубиків 1x1x1. В кожному кубі 1x1x1 записано ціле число. Цей куб містить N горизонтальних «рівнів», а кожен рівень у свою чергу складається з M рядків і K стовпчиків.



Необхідно знайти підмасив цього масиву сума чисел якого максимальна. Тобто знайти такі індекси $i_1, j_1, k_1, i_2, j_2, k_2$ $\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \sum_{k=k_1}^{k_2} A[i, j, k]$ що максимальна.

Вхідні дані:

Перший рядок вхідного файлу b.in містить числа N, M, K. Далі слідує N блоків кожен з яких містить M рядків, по K чисел у кожному – елементи масиву А. Блоки відділені один від одного пустим рядком.

Вихідні дані:

Виведіть у вихідний файл b.out єдине число - максимальну суму.

Обмеження:

$1 \leq N, M, K \leq 30$, елементи масиву – цілі числа з проміжку - 1000..1000.

Умови тестування:

До 40 балів отримає рішення яке коректно працює для N=1 або M=1 або K=1.

До 55 балів отримає рішення яке коректно працює для $N * M * K \leq 1000$.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
2 2 2 7 2 2 3 -1 3 4 6	26
1 3 3 10 4 -1 -2 10 5 5 -3 10	38

Сортування

Задача 1. «Сортування чотирьох»

(V обласна олімпіада, 1991 рік)

Таблиця $a[1:k]$ складається з чисел a, b, c, d , які можуть повторюватись. Відсортувати цю таблицю так, щоб спочатку були записані всі числа d , потім c , потім b і лише потім a . Додаткову таблицю не використовувати.

Задача 2. «Об'єднання таблиць»

(V обласна олімпіада, 1991 рік)

Дано дві впорядковані за зростанням таблиці $A[1:m]$, $B[1:n]$. Скласти алгоритм побудови з цих таблиць впорядкованої за зростанням таблиці $C[1:k]$, $k = m + n$.

Задача 3.

(Заочна олімпіада, 2000-2001)

Дано дійсне число x (не більше 10 знаків разом з десятковою крапкою), натуральне $n \leq 1000$ та елементи a_0, a_1, \dots, a_n ($-30000 \leq a_i \leq 30000$).

Використовуючи дані елементи можна отримати поліном:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0.$$

Пересортуйте елементи a_0, a_1, \dots, a_n так, щоб поліном набув максимального значення.

Вхідні дані: в першому рядку текстового файлу POLINOM.DAT записане дійсне число x , в другому - натуральне число N , в наступних $N+1$ рядках коефіцієнти поліному, починаючи з вільного члена a_0 .

Вихідні дані: в текстовий файл POLINOM.OUT виведіть розв'язок в такому форматі:

в $N+1$ рядках коефіцієнти поліному, починаючи з вільного члена a_0 ; далі значення полінома при новому розташуванні коефіцієнтів. Значення полінома необхідно вивести з точністю до 3-х десяткових знаків.

Приклад вхідних даних:

-2.5

3

1

3

3

1

Приклад вихідних даних:

3

1

3

1

13.625

Задача 4. Сортування

(Заочна олімпіада, 2001-2002)

Напишіть програму, яка відсортує за неспаданням N чисел. Відомо, що $1 \leq N \leq 1000000$, а числа цілі, та за абсолютною величиною не перевищують 100.

З клавіатури вводиться число N , та числа для сортування, відокремлені пропусками.

Відсортований список чисел вивести на екран, відокремлюючи числа одне від одного пропуском.

Приклад вхідних даних:

7 3 -4 3 1 6 -77 6

Приклад вихідних даних:

-77 -4 1 3 3 6 6

Задача 5. Добуток

(Заочна олімпіада, 2001-2002)

Дано N цілих чисел у діапазоні $[-10000..10000]$ ($1 \leq N \leq 100$).

Необхідно вибрати ті з них, чий добуток буде максимальним.

Вхідні дані: у першому рядку число N , у другому N чисел.

Вихідні дані: Числа, у довільному порядку, добуток яких буде максимальний.

Приклад:

5

2 -3 7 0 4

Відповідь:

7 4 2

Задача 6. «Послідовність»

(Заочна олімпіада, перша ліга, 2005-2006)

У файлі **Z3.DAT** записана послідовність двоцифрових натуральних чисел відокремлених пропусками. Довжина послідовності невідома, але не перевищує 1000000.

Напишіть програму, яка запише всі числа, що входять у послідовність в файл **Z3.SOL** у неспадаючому порядку.

Приклад вхідних і вихідних файлів:

Вхідний файл Z3.DAT: 45 56 78 45

Вихідний файл Z3.SOL: 45 45 56 78

Час на тест: 3с

Задача 7. «Калькулятор»

(Заочна олімпіада, вища ліга, 2006-2007)

Калькулятор в лабораторії вміє виконувати операції додавання і множення. Для цього калькулятора створена програма, на вхід якої подається N цілих чисел і працює вона в три етапи:

- 1) вхідні числа заносяться в пам'ять і деякі з них групуються в пари;
- 2) кожна пара видаляється з пам'яті і замість неї заносяться добуток чисел цієї пари;
- 3) знаходиться сума всіх чисел, які знаходяться в пам'яті.

При цьому перший етап виконується так, щоб отримана в результаті сума була максимально можливою. Напишіть програму, яка імітує роботу такого калькулятора.

Вхідні дані. В першому рядку файлу **calc.dat** знаходиться число **N** ($1 \leq N \leq 100$). В другому рядку **N** цілих чисел, кожне з яких за модулем не перевищує 100, числа можуть повторюватись.

Вихідні дані. В єдиний рядок файлу **calc.sol** виведіть одне число - результат виконання програми на калькуляторі.

Приклад вхідних і вихідних даних:

calc.dat

5

0 1 2 3 4

calc.sol

15

Примітка: $15=0+1+2+(3*4)$

Структури даних

Задача 1. «Площа земельної ділянки»

(Заочна олімпіада, 2009-2010)

Фермер Дієтенко вирішив безкоштовно (за поданням уряду) приватизувати землю, для цього йому потрібно обрахувати площу земельної ділянки, яка є багатокутником. Допоможіть фермеру обчислити площу. Для цього він виміряв координати кожної вершини та записав їх у порядку обходу.

Обмеження: $3 \leq N \leq 50000$. Координати по модулю не перевищують 20000. Час 1с. Сторони багатокутника без само перетинів та не мають спільних точок (окрім сусідніх – у вершинах)

Вхідні дані: у текстовому файлі `square.in` у першому рядку записане число N – кількість вершин багатокутника, а у наступних N рядках пари чисел – координати вершини.

Вихідні дані: у текстовий файл `square.out` записати єдине число – площу земельної ділянки. Результат вивести з одним знаком після коми.

Приклад 1:

Вхідний файл: square.in

4

5 0

0 5

-5 0

0 -5

Вихідний файл: square.out

50.0

Теорія графів

Задача 1.

(III обласна олімпіада, 1989 рік)

Схема дорожого між населеними пунктами представлена цілочисленною таблицею $A [1: n, 1: m]$. Елемент таблиці $A [i, j] = 1$, якщо існує дорога між i -м і j -м населеними пунктами, і $A [i, j] = 0$, якщо дороги немає. Скласти алгоритм, який перевіряє чи існує хоча б два населених пункти між якими немає шляху (проходить, бути може, через кілька інших населених пунктів).

Задача 2. «Мережа»

(XI обласна олімпіада, 1997 рік)

Комп'ютерна мережа з'єднує комп'ютери в різних місцях. Не всі комп'ютери з'єднані безпосередньо, але кожен комп'ютер може надсилати повідомлення довільному іншому, передаючи їх через нуль або більше проміжних комп'ютерів. Мережа є деревом, тому завжди існує лише один шлях між деякою парою комп'ютерів.

Проходження повідомлення від одного комп'ютера до іншого, безпосередньо з ним з'єданого, потребує одну секунду (комп'ютери розташовані досить далеко один від одного), і комп'ютер не може відсилати повідомлення кільком комп'ютерам одночасно.

Протягом першої секунди початковий комп'ютер може надіслати повідомлення одному з своїх сусідів. Протягом другої секунди вже два комп'ютери (початковий та його сусід) можуть надіслати повідомлення своїм сусідам, і так далі.

Завдання: Визначити мінімальний час, за який можна передати повідомлення від вказаного комп'ютера всім іншим комп'ютерам в мережі.

Вхідні дані: Опис мережі міститься в файлі **NET.DAT** і має такий формат:

рядок 1: кількість N комп'ютерів в мережі ($1 \leq N \leq 100$)

рядок 2: номер початкового комп'ютера

рядок 3: кількість сусідів комп'ютера 1, потім їх номери, відокремлені пропусками

...

рядок $N+2$: кількість сусідів комп'ютера N , потім їх номери, відокремлені пропусками

Вихідні дані: Результат треба вивести до файлу NET.SOL.

Задача 3. «Розклад маршрутів»

(XII обласна олімпіада, 1998 рік)

Дано розклад всіх автобусних маршрутів міста. Мережу міста треба доїхати з деякої станції A до станції B за найменший час. Складіть програму BUS.*, що визначить найменший необхідний час подорожі. Їхати можна з пересадками.

Задача 4. «Інкасатори»

(XVIII обласна олімпіада, 2004 рік)

Коли Максим Розуменко закінчив фінансовий коледж, він став керуючим міського банку. Вже з перших днів роботи Максим зіткнувся з однією нерозв'язаною для нього проблемою.

В країні, де живе Максим, є N міст. Деякі з них зв'язані двохсторонніми дорогами, які перетинаються тільки в містах. Раз на місяць інкасатори Максима повинні доставити гроші в K зберігальних кас. Всі K зберігальних кас знаходяться в різних містах. Поки банк Максима небагатий і має всього одну машину для перевезення грошей. Вам необхідно допомогти Максиму скласти маршрут, який починається в місті L (в цьому місті знаходиться банк Максима), проходить по всіх K містах, де знаходяться потрібні зберігальні каси, і закінчується також в місті L . Цей маршрут повинен мати мінімальну довжину (довжиною маршруту назвемо суму довжин всіх доріг, що входять в цей маршрут).

Вхідні дані: input.txt

В першому рядку через пропуск записані три числа: N M K, де N – загальне число міст в країні, M - загальне число двохсторонніх доріг, K - кількість зберігальних кас, які повинні відвідати інкасатори (без врахування міста L).

В другому рядку через пропуск йдуть K чисел - номери міст, де знаходяться зберкаси.

Наступні M рядків містять опис доріг. В кожному рядку описується одна дорога у виді X Y S, де X,Y - номери міст, які зв'язані дорогою і S - довжина цієї дороги.

В останньому рядку дано число L - номер міста, де знаходиться банк Максима.

Вихідні данні: output.txt

Єдиний рядок вихідного файлу повинен містити набір чисел - номери міст шуканого маршруту в порядку обходу. Ці числа повинні бути розділені одинарним пропуском. У випадку, коли такого маршруту не існує, вихідний файл повинен містити одне слово "NO".

Обмеження.

$$1 < N \leq 100$$

$$0 < M \leq N(N-1)/2$$

$$0 < K < 18, K < N$$

$$0 < S \leq 50000$$

$$1 \leq L, X, Y \leq N$$

Всі вхідні данні - натуральні числа.

input.txt	output.txt	input.txt	output.txt
4 6 3	1 3 2 4 1	6 9 2	1 2 6 4 3 5 2 1
2 4 3		3 4	
1 2 10		1 3 24	
2 4 3		1 2 10	
1 3 5		2 6 5	

1 4 4	2 5 6
3 2 7	3 5 7
4 3 8	3 4 16
1	4 5 9
	4 6 8
	5 6 4
	1

Задача 5. «Найвіддаленіші міста»

(XIX обласна олімпіада, 2005 рік)

В країні Олімпії є N міст. Деякі міста з'єднані дорогами, які мають певну довжину і по яких можна рухатись в обидва боки. Особливість країни полягає у тому, що з довільного міста можна проїхати у будь-яке інше. Крім того між будь-якими двома містами існує єдиний шлях, який не проходить двічі через одне місто. Мандрівники здавна хотіли взнати, між якими двома містами Олімпії найбільша відстань.

Формат вхідних даних: В першому рядку файлу `cities.dat` знаходиться число N ($2 \leq N \leq 1000$). В наступних рядках по три числа - i, j, s . $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq s \leq 1000$. (Довжина прямої дороги між містами i та j дорівнює s).

Формат вихідних даних: В файл `cities.sol` потрібно вивести відстань між двома найвіддаленішими одне від одного містами.

Приклад вхідних і вихідних даних.

`cities.dat`

5

1 2 10

3 2 2

2 4 5

3 5 2

`cities.sol`

15

Задача 6. «Залізниця»

(XXII обласна олімпіада, 2008 рік)

Залізниця в місті N-ську є замкненою ламаною, яка не має перетинів. Периметр залізниці дорівнює L км. На залізниці розміщені K станцій, пронумерованих від 1 до K за годинниковою стрілкою. Відстань між всіма парами сусідніх станцій рівні. Потяг знаходиться на першій станції і їде з постійною швидкістю V км/год за годинниковою стрілкою. Визначити, на якій станції буде знаходитися потяг через T годин.

Вхідні дані.

Ваша програма повинна вводити з клавіатури числа L , K , V , T у вказаному порядку. Кожне число буде вводитися в окремому рядку.

Вихідні дані.

Ваша програма повинна вивести на екран одне число – номер станції, на якій буде знаходитися потяг через T годин.

Приклад.

Введення	Виведення
20	4
5	
6	
2	

Обмеження.

L , K , V , T – цілі числа; $3 \leq L$, $K \leq 10000$; $0 \leq V$, $T \leq 10000$; гарантується, що відстань між любою парою сусідніх станцій виражається цілим числом кілометрів; гарантується, що через T годин потяг буде знаходитися на деякій станції (а не між станціями).

Задача 7. «Escape»

(XXIV обласна олімпіада, 2010 рік)

Після того, як всі дізналися, що зробив Thunder, йому нічого не залишалось, як втікати від правосуддя. Відомо, що він знайшов в Інтернеті

детальну карту доріг з сусідніми містами. Враховуючи те, що в нього не залишилося грошей він буде переміщуватись по оптимальному шляху. Так, як стало відомо, що ви допомагали йому у злочині, ви повинні допомогти його спіймати. Для цього ви повинні порахувати найменшу вартість переміщення від заданого міста до всіх інших, щоб можна було швидше визначити його маршрут. Так як наші дороги відрізняються якістю, то витрата ресурсів при переміщенні в різних напрямках різна.

Задача 8. Вірус

(Заочна олімпіада, 2001-2002)

Останнім часом було виявлено комп'ютерний вірус, який розповсюджується електронною поштою. Він заснован на такій ідеї: Коли адресат намагається прочитати заражений лист, вірус використовує "дірку" у популярній поштової програмі MacroHard NotLookHere 5.0 і інфікує комп'ютер. На слідує день він розсилає заражені листи всім знайомим власника даного комп'ютера. Маленькій програміст Вася хоче зашкодити проведенню Хмельницької обласної заочної олімпіади, інфікувавши вірусом комп'ютери всіх учасників і журі. Але він може надіслати інфікованого листа тільки одній людині. Напишіть програму, яка визначить кому потрібно надіслати листа, щоб зараження усіх комп'ютерів відбулося якнайшвидше.

Вхідні дані:

Файл INPUT.DAT у першому рядку містить число N ($1 \leq N \leq 100$), у кожному з наступних N рядків по одному імені файлу.

Кожен з цих N файлів описує одного з учасників і має таку структуру:

у першому рядку ім'я учасника,

у другому - його E-mail адреса,

у третьому число P - кількість знайомих даного учасника,

у наступних P рядках містяться імена знайомих.

Вихідні дані: вивести на екран у першому рядку мінімальну кількість днів, яка потрібна для зараження всіх комп'ютерів, у другому рядку вивести E-mail адресу, на яку потрібно відіслати зараженого листа.

Приклад:

Файл INPUT.DAT

4

andrey.dat

vasia.txt

lamer.in

single.txt

Файл ANDREY.DAT

Andrey

andr@myla.net

1

Doctor LAMER

Файл VASIA.TXT

Super Vasia

super@vasia.com

2

Doctor lamer

ANDREY

Файл LAMER.IN

Doctor laMer

lamer@pingvin.fr

3

aNdReY

SUPER Vasia

SINGle

Файл SINGLE.TXT

single

iknow@nobody.com

0

Відповідь:

2

lamer@pingvin.fr

Примітка:

Ім'я учасника, та E-mail адреса - рядок не більш як з 30 символів. Існує хоча б один учасник, від якого вірус за певну кількість днів може розповсюдитись по всім комп'ютерам.

Задача 9. «Гетьманська реформа»

(Заочна олімпіада, 2003-2004)

Біля Чигирину, у степу широкім,
Там могила скіфа, воля козака,
Вони пам'ятають червоні жупани,
Чують голос Гонти і Залізняка...

Щойно обраний гетьман вирішив здійснити адміністративну реформу на Україні. Оскільки війни у ті часи точилися майже без перестану, головним завданням було обрати столицю так, щоб вона була на максимально можливій віддалі від небезпечних територій. На території гетьманської України того часу знаходилось $N \leq 100$, великих вузлових пунктів(міст,селищ). (Назва пункту - послідовність великих латинських літер і символа "-" дефіс, мінус. Кількість символів в назві не більше 30).

Вхідні дані:

У першому рядку два числа N , K та M .

Далі K рядків - назви небезпечних пунктів.

Далі M рядків, кожен з яких містить назву першого пункту, назву другого пункту, відстань між ними.

Вихідні дані:

У перший рядок вивести назву столиці, у другий відстань до найближчого небезпечного пункту.

Приклад вхідних даних:

7 2 7

KONOTOP

GUSIATYN

GUSIATYN KAMIANETS-PODILSKY 1

KAMIANETS-PODILSKY KANIV 3

KONOTOP KYIV 8

KANIV KYIV 2

KANIV BATURYN 1

KANIV CHYGYRYN 3

BATURYN CHYGYRYN 2

Відповідь до прикладу вхідних даних:

CHYGYRYN

7

Задача 10. «Game»

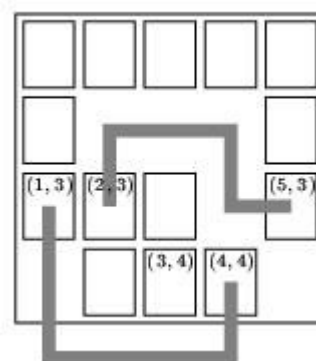
(Заочна олімпіада, вища ліга, 2005-2006)

Я пропоную вам стати одним із розробників нової комп'ютерної гри. Події гри будуть відбуватися на прямокутній дошці розміром $W \times H$. Кожна клітка дошки може містити фішку або не містити нічого.

Особливістю даної віртуальної гри є перевірка того, чи з'єднанні фішки між собою шляхом, що складається з вертикальних та горизонтальних ділянок та не проходить через клітинки, що містять фішки.

Приклад можливих шляхів може бути таким:

Фішки з координатами (1,3) та (4,4) також мають з'єднання.



Вам треба написати програму, яка перевіряє, чи можна з'єднати дві фішки описаним чином і виводить довжину шляху із розрахунку, що шлях між сусідніми клітинками дорівнює 1. Якщо такого шляху не існує – вивести 0.

Вхідні дані:

Перший рядок вхідного файлу **Game.in** містить два натуральні числа: W – ширина дошки, H – висота ($1 \leq W, H \leq 75$). Наступні H рядків містять опис дошки: кожен рядок складається із W символів: символ "X" (велика англійська буква "екс") позначає фішку, символ "." позначає порожню клітинку. Решту рядків містять запити: кожен складається з чотирьох натуральних чисел розділеними пропусками – координат першої та другої фішок. Координати не співпадають. Останній рядок містить чотири 0 – цей запит обробляти не треба, він вказує на кінець вводу. Кількість запитів не більше 20.

Вихідні дані:

Для кожного запиту необхідно вивести одне ціле число в окремому рядку – довжину найкоротшого шляху.

Наприклад:

game.in	game.out
5 4	5
XXXXX	6
X...X	0
XXX.X	
.XXX.	
2 3 5 3	
1 3 4 4 2 3 3 4	
0 0 0 0	
4 4	4
XXXX	

.X..	
..X.	
X...	
1 1 3 1	
0 0 0 0	

Задача 11. «Мудрець та шахи»

(Заочна олімпіада, вища ліга, 2005-2006)

Про мудреця, що створив гру в шахи відомо багато легенд. За однією з них мудрець попросив у царя зерна. Стільки, щоб на першій клітинці шахової дошки була одна зернина, на другій — дві, на третій — чотири. І так в два рази більше зернин на кожній наступній клітинці дошки, аж до 64-ї. Цар пообіцяв мудрецеві таку винагороду, але не зміг виконати своєї обіцянки.

Але ця легенда має й інший фінал.

— Надто ти жадібний мудрець, — розсміявся цар. — Адже у всій моїй країні не знайдеться стільки зерна. Ти мабуть знущаєшся з мене. Але я запропоную тобі іншу винагороду. Постав коня на шахову дошку. І я покладу на цю клітину 128 монет. На кожну клітину, якої кінь зможе досягти за один хід, я покладу 64 монети. На кожну клітину, щоб досягти якої коню потрібно щонайменше два ходи, я покладу 32 монети. А так далі для кожної клітини дошки, відповідно до найменшої кількості ходів конем, що є необхідними, щоб потрапити і відповідну клітину. За кожен хід кількість монет буде зменшуватися в два рази. Але якщо ти виявишся надто жадібним, і не зможеш понести всі монети, які отримаєш, то я накажу відрубати тобі голову.

Мудрець не був фізично сильною людиною, і знав, що не зможе понести більш, ніж 1600 монет. Але він був мудрою людиною, і визначив, на яку клітину слід поставити коня, щоб отримати найбільшу кількість монет, що не перевищує 1600.

Завдання: дано два цілих числа, N та M . Де $64 \leq N \leq 2048$, та N є степенем двійки — кількість монет, яку цар покладе на клітину з конем; $700 \leq M \leq 40000$ — кількість монет, яку може понести мудрець. Напишіть програму, що за даними N та M визначає всі клітинки шахової дошки, на які мудрець може поставити коня, щоб отримати найбільшу кількість монет, що не перевищує M .

Формат вхідних даних: у першому рядку вхідного потоку міститься два числа N та M розділені одним або більше пропусками.

Формат вихідних даних: у першу стрічку вихідного потоку вивести всі клітини, що відповідають умові задачі, у вигляді, прийнятому для позначення клітин шахової дошки, розділивши клітини одним або більше пропусками. (Клітини шахової дошки позначаються парами символів. Перший символ — мала латинська літера, другий символ цифра).

Приклад вхідних даних:

128 1600

Приклад вихідних даних, що відповідають вхідним:

c2 f2 b3 g3 b6 g6 c7 f7

Задача 12. «Царевич ховається»

(Заочна олімпіада, вища ліга, 2005-2006)

Царевич та його нянька грають в хованки в палаці. Спочатку вони знаходяться в одній кімнаті. Царевич біжить ховатися, а нянька чекає доки він заховається. Всі двері в усіх кімнатах палацу відчинені. Але в царевича є ключі від усіх дверей в палаці. Він біжить з кімнати в кімнату і запирає за собою двері на ключ. Після цього ні він, ні нянька не відчиняють запертих дверей. Царевич може кілька разів пробігати через кімнату, де залишилася нянька, якщо це дозволяють ще не заперті двері. Царевич ховається в якійсь кімнаті, і чекає поки нянька його знайде. Після того, як царевич заховався, нянька вирушає його шукати, і знаходить, якщо вона може потрапити в кімнату, де заховався царевич.

Завдання: напишіть програму, що знаходить шлях, який повинен пробігти царевич по кімнатах палацу, щоб нянька не змогла його знайти.

Формат вхідних даних: в першій стрічці вхідного потоку міститься три цілих числа N , M та D . $2 \leq N \leq 30$ — кількість кімнат в палаці; $1 \leq M \leq N$ — номер кімнати, де знаходяться царевич та його нянька; D — кількість дверей. В наступних D стрічках вхідного потоку знаходяться пари цілих чисел $1 \leq a \neq b \leq N$ — номери кімнат, між якими є двері. Будь-яка пара кімнат з'єднана не більш, ніж одними дверима.

Формат вихідних даних: в першу стрічку вихідного потоку вивести одне число K — кількість кімнат, що їх має пробігти царевич, щоб заховатися, враховуючи початкову кімнату, та кімнату, де він захищається. В наступну стрічку вихідного потоку вивести K цілих чисел, розділених одним або більше пропусками — номери кімнат, через які має пробігти царевич. Якщо за даних умов царевич не може заховатися так, щоб нянька його не знайшла, то в першу стрічку вихідного потоку вивести єдине число 0.

Приклад вхідних даних:

```
4 1 5
1 2
1 4
2 3
2 4
3 4
```

Приклад вихідних даних, що відповідають вхідним:

```
4
1 4 2 3
```

Задача 13. «Переслідування злочинця»

(Заочна олімпіада, перша ліга, 2006-2007)

В країні Програмія є N міст. Деякі пари міст з'єднані двосторонньою дорогою. За даними поліції з міста з номером A у місто з номером B вирушив небезпечний злочинець. Для затримання цього злочинця поліція вирішила

перекрити шляхи до міста **B**. Але через обмежений бюджет і значні витрати на перекриття дороги вони намагаються, щоб кількість перекритих доріг була мінімальною і при цьому з міста **A** до міста **B** було неможливо добратись.

Вхідні дані: в першому рядку файлу **chase.dat** записані через пробіл числа **N**, **K**, **A**, **B** ($2 \leq N \leq 50$, $1 \leq A$, $B \leq N$). В наступних **K** рядках знаходяться по два числа - пари міст, між якими існує дорога.

Вихідні дані: в файл **chase.sol** потрібно вивести одне число - мінімальну кількість доріг, які потрібно перекрити, щоб в результаті не було вільного шляху від **A** до **B**.

Приклад вхідних і вихідних даних:

chase.dat

5 6 2 5

1 2

1 5

2 3

2 4

3 4

4 5

chase.sol

2

Задача 14. «Проблема «Синіх» та «Помаранчевих»

(Заочна олімпіада, вища ліга, 2006-2007)

На території деякої держави ідуть військові навчальні маневри між двома сторонами: «Синіми» і «Помаранчевими». Особливості ландшафту і складні кліматичні умови вимагають підрозділи обох сторін розташовуватись тільки на території деяких населених пунктів. Загальна кількість населених пунктів у державі рівна **N**.

Тактика ведення бойових дій «Помаранчевих» розрахована на нанесення супротивнику швидких і раптових ударів, що можливо лише у

тому випадку, якщо у операціях задіяні моторизовані частини, а їхнє пересування відбувається тільки по дорогах.

Розмаїтість використовуваної бойової техніки призводить до того, що час пересування різних бойових частин із одного пункту до іншого різний, і визначається величиною V_j - швидкістю руху підрозділів військової частини розквартированої у j -му населеному пункті.

Використовуючи величезну перевагу у техніці, «Помаранчеві» планують організувати нічний наліт на бази супротивника і повністю їх розгромити. Всі військові підрозділи «Помаранчевих» приступають до виконання операції одночасно. Якщо бойова частина «Помаранчевих» вривається у населений пункт, зайнятий «Синіми», то враховуючи фактор раптовості, їм вдається повністю розгромити це угруповання.

Близькучому проведенню цієї операції завадило те, що через деякий час T після початку операції «Синіми» було здійснено радіоперехоплення повідомлення про операцію, що розпочалася. Після радіоперехоплення угруповання «Синіх» миттєво розсіюються в околиць горах і залишаються непошкодженими.

Вияснить, яку кількість угруповань супротивника і в яких населених пунктах «Помаранчевим» все-таки вдасться розгромити «Синіх».

Передбачено, що у початковий момент часу угруповання «Помаранчевих» і «Синіх» не можуть знаходитись у одному і тому ж населеному пункті. Якщо сигнал тривоги надходить у той момент, коли бойова частина «Помаранчевих» тільки вривається у населений пункт, зайнятий «Синіми», то, використовуючи чудове знання місцевості, «Синім» все-таки вдається сховатися в горах. Величезна перевага у техніці і живій силі дозволяє бойовим частинам «Помаранчевих» організувати із кожної частини довільну кількість експедицій для розгрому «Синіх». Ніщо не заважає одній експедиції за час проведення операції знищити кілька угруповань.

Формат вхідних даних: вхідні дані задачі записані у файлі **victory.in**.

У першому рядку цього файлу записані два числа: **N** – кількість населених пунктів ($1 \leq N \leq 200$) і **K** – кількість доріг, що з'єднують ці населені пункти ($0 \leq K \leq 1000$). Дороги ніде не перетинаються. У наступних **K** рядках файлу міститься схема доріг. У кожному рядку записано два натуральних числа **i** та **j** і одне додатне дійсне число **L_{ij}**, яке означає, що між населеними пунктами **i** та **j** існує дорога довжиною **L_{ij}** кілометрів ($L_{ij} < 1000$). Далі описується розміщення бойових частин сторін, що воюють. Наступний рядок файлу містить число **M₁** – кількість бойових частин «Помаранчевих». У наступних **M₁** рядках записано по два числа: перше – ціле число **j** – номер населеного пункту, у якому розміщується військова частина; друге – дійсне невід'ємне число **V_j** – швидкість руху бойових колон цієї частини у кілометрах за годину ($V_j < 110$). Далі у окремому рядку файлу записано число **M₂** – кількість бойових угруповань «Синіх», за яким записані **M₂** чисел – номери населених пунктів, де розташовані ці угруповання. У останньому рядку файлу записане одне додатне дійсне число **T**, виміряне у годинах ($T < 24$). Всі числа розділені пропусками.

Формат вихідних даних: у перший рядок файлу **victory.out** виведіть кількість розгромлених угруповань, а у другий – номери населених пунктів (у порядку зростання), у яких ці угруповання базувались.

Приклад введення і виведення:

victory.in	victory.out
8 7	2
1 2 80	4 8
2 4 25	
4 5 10	
6 2 5	
2 3 40	
7 6 10	

8 7 15	
2	
1 50	
6 20	
4 4 5 3 8	
2.0	

Задача 15. «Метаморфози»

(Заочна олімпіада, вища ліга, 2006-2007)

Вчитель олімпійця Сашка з першої задачі, виявивши, що до його комп'ютера був здійснений несанкціонований доступ, з переляку написав доповідну директору школи, а той в свою чергу направив його доповідну до Служби Безпеки України. Дана справа про несанкціонований доступ до інформаційної системи була передана полковнику Іваненку з другої задачі. Він швидко зібрав інформацію по школі, і зробив висновок, що в школі існують лише дві особи, які здатні це зробити: сам вчитель і олімпієць Сашко (решта не вміють вмикати комп'ютери).

Отже, підозра впала на Сашка, і він вирішив втікти в інше місто до свого друга олімпійця Миколи. Однокласник Вася (для якого Сашко вкрав коди доступу) дав йому свій велосипед. Сашко вирішив якнайшвидше дістатись кінцевого пункту. Він взяв свій ноутбук, в якому була база даних, що містила карту України. У Сашка була вся інформація щодо того, скільки часу займає його пересування між різними парами міст, і скільки часу він повинен витратити, щоб переїхати кожне місто. Необхідно визначити найменший час. За який Сашко приїде до Миколи.

Вхідні дані:

В першому рядку вхідного потоку містяться числа **N** та **M** – кількість міст та кількість доріг.

В кожному з наступних M рядків міститься по 3 цілих числа A, B, C – номери міст, які сполучає дорога і час який необхідно подолати при переїзді з міста A в місто B (і навпаки). ($1 \leq C \leq 100$)

В останньому рядку міститься N цілих чисел – час переїзду через відповідні міста, всі числа з цього рядка лежать в межах від $1..100$.

Сашко знаходиться в місті номер 1 , а Микола мешкає у місті номер N . ($1 < N \leq 200$)

Вихідні дані:

У вихідний потік виведіть число – мінімальний час, за який Сашко може дістатися до Миколи (відповідний шлях є завжди)

Приклад вхідних даних:

```
4 5
1 2 2
1 3 3
2 4 2
2 3 3
3 4 20
7 1 10 8
```

Приклад вихідних даних:

```
20
```

Задача 16. «Tour De France»

(Заочна олімпіада, 2008-2009)

Ім'я вхідного файлу:

```
tour.in
```

Ім'я вихідного файлу:

```
tour.out
```

На цей раз Вам доведеться допомогти організаторам всесвітньо відомих велоперегонів Tour De France виготовити трасу, по якій буде проходити змагання. Організатори перегонів надали вам карту. На ній

відмічено N пунктів і M одно-направлених доріг, які з'єднують ці пункти. Слід зауважити, що транспортна система в Європі передбачає, що між будь-якою парою пунктів буде існувати як мінімум одна дорога (про її направленість не повідомляється). Траса повинна починатися з будь-якого пункту і має бути прокладена через всі N пунктів, а також має слідувати $N-1$ дорогами (які з'єднують відповідні N пунктів).

Формат вхідних даних: у першому рядку вхідного файлу записано два числа N ($1 \leq N \leq 100$) та M ($1 \leq M \leq 10000$). В наступних M рядках записано по два числа a_i та b_i ($1 \leq a_i, b_i \leq N$), що задають дорогу, яка виходить з пункту a_i й слідує до пункту b_i . Номери пунктів – натуральні числа, що не перевищують N . Причому, можуть існувати дороги, які виходять і входять в один і той самий пункт, а також два пункти, можуть бути з'єднані більше ніж однією дорогою. Однак, гарантується, що для будь-яких двох різних пунктів з номерами i та j існує дорога, яка виходить з пункту i і слідує до пункту j , або ж навпаки – виходить з пункту j та слідує до пункту i .

Формат вихідних даних: у випадку існування такої траси у єдиному рядку вихідного файлу має бути записано N чисел – номери пунктів, по яким буде проходити траса, в порядку її слідування, відокремлених рівно одним пробілом. Якщо існує декілька варіантів прокладки траси – виведіть будь-яку. Якщо варіантів прокладання траси не існує, то в єдиному рядку вихідного файлу виведіть рядок "NO SOLUTION", без лапок.

Приклади вхідних та вихідних даних:

tour.in

4 6

1 2

2 3

1 4

4 3

4 2

1 3

tour.out

1 4 2 3

Задача 17. «Прогулянка по місту»

(Заочна олімпіада, 2009-2010)

Ім'я вхідного файлу:

travel.in

Ім'я вихідного файлу:

travel.out

Максимальний час роботи на одному тесті:

800 ms

Максимальна оцінка за задачу:

50 балів

У Кракові, недалеко від берега Вісли стоїть Дракон: кам'яний, з пухирчастим тілом, величезними лапами, до неба піднятою головою. Місто Краків – улюблене місто туристів, які з'їжджаються сюди звідусіль. Петрик – один із них. Карта Кракова являє собою таблицю, розміром $N \times M$, клітинками якої є квартали, деякі з яких доступні для руху, а деякі ні. Також на карті відмічені цікаві місця, причому це також квартали, по яких можливий рух. Вася вибрав маршрут для ознайомлення з містом. Він почне з клітинки (1;1), прийде в клітинку (N;M), а потім знову повернеться туди звідки прийшов (в клітинку (1;1)). Звичайно, Петрик хоче переглянути якнайбільше різних цікавих місць. Так як Петрик знає Краків дуже погано, та й взагалі заблукати тут дуже небажано, він вирішив, що на шляху до клітинки (N;M) буде завжди іти вправо, або вниз (відносно карти), а на зворотнім – вліво, або вверх. Допоможіть Петрикові знайти максимальну кількість різних цікавих місць, в яких він може побувати.

Формат вхідних даних: у першому рядку вхідного файлу міститься два числа – N та M ($1 \leq N, M \leq 100$). В наступних N рядках міститься опис карти – рядок з M літер, причому “#” означає квартал недоступний для руху, “*” –

місце для перегляду, а “.” (точка) – квартал доступний для руху, який слугує Вамі лише клітинкою для руху. Гарантується, що клітинки (1;1) й (N;M) не будуть позначені символом “#”.

Формат вихідних даних: у єдиному рядку вихідного файлу повинно міститися одне число – максимальна кількість різних місць для перегляду, в яких може побувати Петрик, протягом свого маршруту, або -1, якщо маршрут здійснити не вдасться.

Задача 18. «Труба часу»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Труба часу – це однобічний тунель, що сполучає дві зоряні системи. Час руху по тунелю миттєво, але при цьому можна потрапити або в майбутнє, або в минуле на певне число років. Тунелі сполучають лише різні зоряні системи. Відомо, що з Сонячної системи можна потрапити в будь-яку іншу зоряну систему. У завданні необхідно встановити, чи можна землянинові, рухаючись по трубах часу, потрапити в безконечне минуле.

Вхідні дані:

Перший рядок містить кількість тестів k (k - не перевищує 10). Перший рядок кожного тесту містить число зоряних систем n ($1 \leq n \leq 1000$) і число труб часу m ($1 \leq m \leq 2000$). Зоряні системи нумеруються числами від 0 до $n - 1$ (Сонячна система має номер 0). Наступні m рядків містять три числа x, y, t ($-1000 \leq t \leq 1000$). При переміщенні із зоряної системи x в систему y відбувається завдяки переміщенню в часі на t років. Якщо $t > 0$, то переміщення відбувається в майбутнє, якщо $t < 0$ – то в минуле.

Вихідні дані:

Для кожного тесту вивести повідомлення "Yes" або "No" залежно від можливості або неможливості потрапити в нескінчене минуле.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
2	
3 3	
0 1 1000	Yes
1 2 15	No
2 1 -42	
4 4	
0 1 10	
1 2 20	
2 3 30	
3 0 -60	

Задача 19. «Слонові походи»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 0.5 с

Ліміт пам'яті: 64 Мбт

Василько любив грати в шахи і часто придумував для себе різні шахові головоломки. Ось одна із них. Нехай є шахова дошка розміром $N \times N$ ($1 \leq N \leq 10^9$). Потрібно знайти скільки ходів треба слону, щоб з клітинки (x_1, y_1) потрапити в клітинку (x_2, y_2) .

Вхідні дані:

У першому рядку стандартного вхідного потоку знаходиться натуральне T ($1 \leq T \leq 100$) - кількість тестів. Кожен тест містить 5 цілих чисел: N, x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq N$).

Вихідні дані:

Для кожного тесту вивести кількість ходів слона або "no move" для випадку, коли слон не може потрапити з першої клітинки в другу.

Приклад вхідних та вихідних даних.

Приклад даних:	вхідних	Приклад даних:	вихідних
3		1	
8 3 6 6 3		no move	
8 4 2 2 3		2	
6 1 4 6 5			

Задача 20. «Кодекс»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

В деякій країні парламент вирішив прийняти інформаційний кодекс, яким дуже сильно обмежував використання інтернету. Відтепер перевіряючий зможе коли завгодно читати чужу електронну пошту, на відвідування сайтів необхідно з'явити дозволу, і крім того користувачам становлено річний ліміт завантаженої інформації 1 Гб. Звісно це дуже обурило користувачів інтернету, і вони вийшли на мітинг, мітингували довго і нудно, і зрозуміли, що необхідно змінити тактику боротьби за свободу Інтернета. В координаційному штабі інтернетчиків визрів план цифрового звільнення, що передбачав хакерський взлом комп'ютерів парламенту. Як відомо комп'ютери в парламенті цієї країни об'єднані в специфічну мережу з наступними характеристиками:

- 1) Кожен комп'ютер з'єднаний з певною кількістю інших, і може передати їм інформацію;
- 2) Між будь-якими двома комп'ютерами є лише 1 шлях передачі інформації;
- 3) Отримавши доступ до комп'ютера можна з'явити під контроль його і всі комп'ютери з якими він безпосередньо з'єднаний.

Задача координаційного штабу інтернетчиків, визначити найменшу кількість комп'ютерів, які треба взламати, аби отримати повний доступ до мережі парламенту.

Вхідні дані:

У першому рядку число N – кількість комп'ютерів в парламенті. У другому рядку число M – кількість міжкомп'ютерних з'єднань. В кожному з наступних M рядків міститься по 2 числа – номери комп'ютерів у з'єднанні. ($1 < N \leq 10000$)

Вихідні дані:

Єдине число – мінімальна кількість комп'ютерів, яку необхідно взламати.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
8	2
7	
1 2	
2 3	
3 4	
4 5	
2 6	
4 7	
4 8	

Теорія ігор

Задача 1.

(III обласна олімпіада, 1989 рік)

У гру "Клац" грають так: поле $M \times N$ заповнено фішками. Два гравці по черзі знімають з поля фішки. Гравець при своєму ході повинен вибрати будь-яку незняту фішку і зняти її разом з усіма фішками, номер рядки яких більше або дорівнює номеру рядка обраної фішки, а номер стовпчика - більше або дорівнює номеру стовпця обраної фішки.

Програє той, хто зніме з поля фішку, що стоїть у лівому нижньому кутку (тобто в клітині $[1,1]$). У кого з гравців є виграшна стратегія при $M=2$? Розробити програму, що грає за цього гравця.

Задача 2.

(III обласна олімпіада, 1989 рік)

Гра "Баше" (у грі беруть участь двоє). Суперники ходять по черзі і за кожний хід будь-якої з граючих може взяти не більше M предметів. Програє той, хто змушений взяти останній предмет. Скласти програму, що грає за гравця з виграшною стратегією.

Задача 3.

(IV обласна олімпіада, 1990 рік)

Двоє людей грають у таку гру: є поле $M \times N$ (M і N - непарні), на яке гравці по черзі ставлять по одному шахового короля. Короля можна ставити на будь-яку клітину, яка не зайнята і не перебуває під боєм якого-небудь короля, поставленого раніше будь-яким з гравців. Гравець, який не зможе зробити черговий хід, вважається що програв.

Написати програму, яка виступає за гравця, що має виграшну стратегію.

Задача 4. «Морський бій»

(VII обласна олімпіада, 1993 рік)

Скласти програму «Морський бій». Гра проходить між двома учасниками, один з яких ЕОМ, на полі 10 x 10 кліток. Кожний з учасників розставляє на полі по 10 однокліткових катерів (катери не можуть дотикатись один до одного). При попаданні в катер при черговому ході він вважається знищеним. Перемагає той, хто знищить першим катери суперника. Якщо учасник під час чергового ходу знищив катер суперника, то він ходить ще раз. ЕОМ розміщує катери випадковим чином, її партнер водить координати катерів з клавіатури.

Додаток. На полі кожний учасник розставляє ескадру, що складається з одного чотирікліткового лінкора, двох трикліткових крейсерів, трьох двокліткових есмінців і чотирьох однокліткових катерів. Всі багатокліткові кораблі розміщені лінійно, тобто тільки горизонтально чи вертикально (Кількість балів збільшується в 1,3 рази).

Задача 5. «Жадібний трикутник»

(XX обласна олімпіада, 2006 рік)

Журі заочної олімпіади кинуло виклик команді олімпійців у грі "Жадібний трикутник".

Правила гри наступні:

В гру грає двоє гравців (команда журі і команда олімпійців). У трикутному полі з N рядків (стовпчиків та діагоналей) записані цілі числа. За один хід дозволяється забрати всі числа або верхнього рядка, або лівого стовпчика, або нижньої діагоналі, після чого залишається трикутник меншого розміру. Гравці виконують ходи по черзі. Гра закінчується, коли не залишиться жодного числа. Виграє учасник, який в результаті набере більшу суму чисел.

Звісно, що журі дуже професійне, тому спробуйте написати програму, яка допоможе вам у нелегкій боротьбі.

Вхідні дані: (файл tr.dat)

У першому рядку записано число N ($1 \leq N \leq 30$). У наступних N рядках записані числа, що становлять "жадібний трикутник". Кожне з цих чисел більше від 0 та не перевищує 50.

Вихідні дані: (файл tr.sol)

У результуючий файл запишіть єдине число - різницю між сумою чисел які збере гравець який почне гру першим і сумою чисел які збере той, хто буде робити свій хід другим (при найкращій грі обох гравців).

Приклад tr.dat	Приклад tr.sol
4 1 2 7 3 4 5 3 2 7 1	9

Задача 6. «Гра з обмінами»

(XXII обласна олімпіада, 2008 рік)

Сашко і Олег грають у захоплюючу гру. Сашко виписав підряд числа від 1 до N . А Олег виписав P пар чисел (A_i, B_i) .

Тепер Сашко перетворює отриману послідовність чисел - він міняє місцями числа в цій послідовності. Якщо деяка пара чисел (A_i, B_i) виписана Олегом, то Сашко має право в будь-який момент взяти числа із послідовності, які стоять на місцях A_i і B_i і поміняти їх місцями.

Наприклад, нехай $N=5$. Тоді спочатку Сашком виписана послідовність
1 2 3 4 5

Нехай Олег написав дві пари чисел: $(1,2)$ і $(2,5)$. Тоді Сашко в любий момент може міняти числа, які стоять на 1 і 2 місцях, або ж числа, які стоять на 2 і 5 місцях.

Наприклад, він може послідовно отримати такі послідовності:

2 1 3 4 5 (поміняв числа на 1 і 2 місцях)

2 5 3 4 1 (поміняв числа на 2 і 5 місцях)

5 2 3 4 1 (поміняв числа на 1 і 2 місцях).

Олегу не показують проміжні послідовності, а записують лише отриману на останньому кроці.

Від Олега вимагається перевірити, чи міг Сашко отримати таку послідовність не порушуючи правил гри, і якщо міг, то вказати, в результаті якої послідовності обмінів (при цьому не вимагається, щоб число обмінів було мінімально можливим).

Напишіть програму, яка допоможе Олегу справитися з цією задачею.

Формат вхідних даних

Спочатку записано число N ($1 \leq N \leq 100$) – кількість чисел в послідовності. Далі йде N чисел – послідовність, яку отримав Сашко (в послідовності кожне з чисел від 1 до N зустрічається рівно один раз).

Далі йде число P ($0 \leq P \leq 10000$) – кількість пар чисел, які виписав Олег. Далі записано P пар чисел (кожне число пари – з діапазону від 1 до N).

Формат вихідних даних

В перший рядок вихідного файлу виведіть повідомлення YES (якщо така послідовність могла бути чесно отримана Сашком) і NO (якщо таку послідовність Сашко не міг отримати, не порушуючи правил гри).

У випадку, коли така послідовність могла бути отримана, далі виведіть спосіб її отримання (якщо варіантів декілька, виведіть будь-який із них). Спочатку виведіть число K – кількість операцій обміну (воно не повинно перевищувати 100000), а в наступні K рядків – K пар чисел, які задають номери місць, на яких стоять елементи, що обмінюються (числа в парі можуть бути видані в будь-якому порядку). Гарантується, що якщо розв'язок існує, то існує розв'язок з числом обмінів, який не перевищує 100000.

Задача 7. Вовк та вівці

(Заочна олімпіада, 2001-2002)

Для цієї гри використовується шахова дошка, одна чорна ("Вовк") та чотири білих ("Вівці") шашки. На початку гри "Вовк" займає клітинку C1, а "Вівці" B8, D8, F8 та H8. Усі ходи дозволені тільки на одну клітинку по

діагонали, при чому "Вовк" може рухатись як вперед, так і назад (4 напрямки), а "Вівці" тільки вперед (2 напрямки). "Вівці" та "Вовк" ходять по чергово. Перший хід роблять "Вівці". За один хід рухається тільки одна "Вівця". Програє той, хто не зможе зробити свій хід. (Або "Вівці" дійдуть протилежного краю дошки, або "Вівці" оточуть "Вовка").

Напишіть програму, яка буде грати проти "Вовка".

Для учасників, що пишуть на Borland Pascal:

Вашій програмі при компіляції буде надано модуль GAME.TPU, в якому будуть реалізовані

такі функції та процедури:

```
PROCEDURE InitGame;
```

```
PROCEDURE MakeMove(S1,S2:string);
```

```
FUNCTION GetMove:string;
```

```
PROCEDURE EndGame;
```

InitGame повинна бути викликана на початку програми.

Для того, щоб повідомити "Вовку" свій хід програма повинна викликати MakeMove, де S1 та S2

початкове та кінцеве положення "Вівці", що робить хід.

Для того, щоб взяти хід "Вовка" програма повинна викликати функцію GetMove, яка поверне

нове положення "Вовка".

Коли настане положення, де одна із сторін не зможе зробити свій хід потрібно викликати

```
EndGame.
```

Ваша програма може мати приблизно таку структуру:

```
USES Game;
```

```
....
```

```
BEGIN
```

```
InitGame;
```

```
....
```

```
repeat
....
MakeMove(S1,S2);
....
S:=GetMove;
....
until ...
EndGame;
END.
```

Для учасників, що пишуть на C++:

Вашій програмі при компіляції буде надано файл GAME.H, в якому будуть реалізовані такі функції:

```
void InitGame();
void MakeMove(char* S1,char* S2);
char* GetMove();
void EndGame();
```

Призначення цих функцій теж саме, як і Pascal аналогів.

Ваша програма може мати приблизно таку структуру:

```
include <game.h>
....
int main()
{
InitGame();
....
while(...)
{
....
MakeMove(S1,S2);
....
S=GetMove();
```

```
....  
}  
EndGame();  
return 0;  
}
```

Примітка.

Кожне положення "Вовка" чи "Вівці" кодується рядком із двох символів: Перше велика латинська літера (A..H), що позначає рядок, друга цифра (1..8), що позначає стовпчик. Отже при першому ході "Вівця" що стоїть на полі B8 може зробити хід або на A7, або на C7.

Програма буде тестуватися двома або трьома "Вовками", кожен з яких буде мати свій рівень складності.

Задача 8.

(Заочна олімпіада, 2002-2003)

В місті Безвиграшне раз в тиждень організують таку інтелектуальну лотерею програмістів:

- 1) протягом тижня програмісти міста телефонують до організаторів лотереї;
- 2) випадковим чином вибираються і запрошуються в студію претендентів на головний приз;
- 3) з кожним претендентом окремо проводять розіграш описаний в пунктах
4) та 5) та 6);
- 4) в лототрон закладають 1000) пронумерованих куль з номерами від 1 до 1000;
- 5) робот GIRL достає 1000 разів випадковим чином кулі з барабана, а програма претендента на виграш вгадує номер кожної кулі;
- 6) доки програма претендента не назве правильно номер кулі, наступна куля не виймається з лототрона.

7) головний приз дістається тому претенденту, програма якого витратить на відгадування всіх номерів найменшу сумарну кількість запитів.

Напишіть програму для портативного комп'ютера гравця, яка буде створювати запити.

Програмістам, що пишуть на мові Pascal буде надано модуль GIRL.TPU,

в якому буде реалізована така функція:

```
FUNCTION Zapyt(Number:Integer):Integer
```

Програмістам, що пишуть на мові C++ буде надана бібліотека GIRL. H, в якій буде реалізована така функція: int zapyt(int Number)

Функція повертає 1, якщо число в запиті більше номера кулі.

Функція повертає -1, якщо число в запиті менше номера кулі.

Функція повертає 0, якщо число в запиті дорівнює номеру кулі.

Наведений нижче приклад правильний, але не дає виграшного результату:

```
USES GIRL;  
VAR i,j : Integer;  
BEGIN  
    For i := 1 to 1000 do  
        For j:= 1 to 1000 do  
            If Zapyt(j)=0 Then Break;  
        END.  
    END.
```

Задача 9. «Новий Рік не прийде»

(Заочна олімпіада, 2002-2003)

Подорожуючи до міста Olympijsk з країни Вічних Снігів, Дід Мороз зупинився у одному з готелів невеличкого містечка. У місцевому ресторані він познайомився з ICE'om - видатним гравцем у "Крижинки", який запропонував зіграти Діду Морозу на його подарунки.

У випадку програшу Діда Мороза, він програє свою чарівну торбинку з подарунками, без якої свято Нового Року не відбудеться. Знаючи, що Дід Мороз не вміє грати у "Крижинки" ІСЕ запропонував йому вирішити, буде він ходити першим чи другим.

Гра відбувається на полі розміру $N \times N$. У першому рядку розташовані крижинки ІСЕ'а, у N -ому крижинки Діда Мороза. На кожному ході гравець обирає свою крижинку та переміщує її вперед на будь-яку кількість вільних клітинок. Програє гравець, що не зможе зробити хід.

$(3 \leq N \leq 1000)$

Приклад:

Початкова позиція при полі 5x5

XXXXX

.....

.....

.....

OOOOO

Можливий хід ІСЕ'а

XX.XX

.....

..X..

.....

OOOOO

Можлива відповідь Діда Мороза

XX.XX

.....

..X..

..O..

OO.OO

Можливий хід ІСЕ'а

X..XX

.....
..X..
.XO..
OO.OO

Напишіть програму, яка буде безпрограшно грати за Діда Мороза.

Вказівки для розв'язків на мові Pascal

При компіляції вашій програмі буде надано модуль GAME.TPU

У цьому модулі буде реалізовано такі функції та процедури:

```
FUNCTION InitGame:integer;  
PROCEDURE BeginGame(K:integer);  
PROCEDURE MakeDidMorozMove(C,P:integer);  
PROCEDURE GetIceMove(var C,P:integer);  
PROCEDURE EndGame;
```

На початку гри викличте функцію InitGame, яка поверне значення N.

Далі ваша програма повинна викликати процедуру BeginGame(K)

де K = 1, якщо перший хід буде робити ICE,

K = 2, якщо перший хід буде робити Дід Мороз.

Далі, при ході Діда Мороза потрібно викликати процедуру MakeDidMorozMove(C,P), де C - номер стовпчика у якому робиться хід, P - кількість клітинок, на яку перемістилась крижинка.

При ході ICE'а потрібно викликати процедуру GetIceMove(C,P).
Значення C та P такі ж, як і в процедурі MakeDidMorozMove(C,P)

У кінці гри (коли жоден з гравців не зможе зробити хід) необхідно викликати процедуру EndGame.

Ваша програма може виглядати таким чином:

```
USES Game;  
...  
BEGIN  
...  
N:=InitGame;
```

```

...
BeginGame(K);
...
while...
begin
...
  if flag then MakeDidMorozMove(C,P)
    else GetIceMove(C,P);
...
end;
EndGame;
END.

```

Вказівки для розв'язків на мові C++

При компіляції вашій програмі буде надано файл GAME.H

У цьому модулі буде реалізовано такі функції:

```

int InitGame();
void BeginGame(int K);
void MakeDidMorozMove(int C, int P);
void GetIceMove(int &C, int &P);
void EndGame();

```

Зміст цих функцій відповідний аналогічним функціям та процедурам з модуля GAME.TPU для розв'язків на мові Pascal.

Можливий вигляд вашої програми:

```

#include <game.h>
...
int main()
{
...
  N=InitGame();

```

```

...
BeginGame(K);
...
while...
{
...
if(flag) MakeDidMorozMove(C,P);
else GetIceMove(C,P);
...
}
EndGame();
return 0;
}

```

Задача 10. «У Таврію за волею...»

(Заочна олімпіада, 2003-2004)

Зібралися одного разу козаки в похід до кримського ханства, дівчат українських з полону визволяти, прийшли оточили ханський палац, та й кажуть: "А ну бусурмани, віддавайте дівчат наших, бо скучили вже сади вишневі за ними, а як не віддасте, то будемо битися з вами, аж поки останній з нас не поляже чи не посічемо вас усіх на капусту...".

Злякався хан шабель козацьких, але й полонянок віддавати не хотів. Надіслав він листа отаману кошовому: "Навіщо нам кров проливати, от як виграєш ти в мене в хрестики-нулики віддам дівчат ваших без бою". І став кошовий думу думати, як хана-бусурмана обіграти, і винайшли козаки шлях до виграшу, а чи ви знайдете?

Хрестики-нулики граються на полі розміром 3x3. Усі 9 клітинок на початку гри порожні. Перший хід має гравець, що ставить хрестики. Гравці по черзі ставлять у порожні клітинки свої знаки (перший - хрестики, другий - нулики). Гра закінчується коли:

А) на певній горизонталі, вертикалі чи діагоналі є три однакових знаки.
Хто зміг виставити свої знаки відповідним чином 3 в ряд, той і виграв.

Б) Коли пункт "А" не здійснився, і не залишилось вільних клітинок.
Результат у цьому випадку - нічия.

Ваша програма повинна:

На початку викликати функцію BeginGame, яка поверне значення 0 чи 1. 0 - ви граєте нуликами, 1 - хрестиками.

Для того, щоб зробити свій хід, викликати процедуру MakeMove(r,c) де параметри r та c - номер рядка та стовпчика в який ви робите хід.

Для того, щоб взяти хід суперника викликати процедуру TakeMove(r,c), яка поверне у змінні r та c номер рядка та стовпчика в який робить хід суперник.

Якщо гра закінчується - викликати процедуру EndGame(res). В параметр res - передати 0 - якщо ви програли, 1 - у випадку нічиєї, 2 - якщо ви виграли. Програма буде перевірятися на декількох модулях, різного рівня складності.

Усі вищенаведені процедури знаходяться у модулі zeros.tpu для мови Pascal (не використовуйте інших модулів! CRT,DOS і т.д.) і у файлі zeros.h для мови C++.

Ваша програма мовою Pascal може мати наступний вигляд:

```
USES zeros;  
.....  
BEGIN  
...  
K:=BeginGame;  
...  
MakeMove(r,c);  
...  
TakeMove(r,c);  
...
```

```
EndGame(res);
```

```
END.
```

Ваша программа мовою C++ може мати наступний вигляд:

```
#include "zeros.h"
```

```
...
```

```
int main()
```

```
{
```

```
    k=BeginGame();
```

```
    ...
```

```
    MakeMove(r,c);
```

```
    ...
```

```
    TakeMove(r,c);
```

```
    ...
```

```
    EndGame(res);
```

```
    return 0;
```

```
}
```

Вам надаються зразки модулів (але грають вони не найкращим чином)

```
{Zeros.pas}
```

```
UNIT zeros;
```

```
INTERFACE
```

```
FUNCTION BeginGame:integer;
```

```
PROCEDURE EndGame(res:integer);
```

```
PROCEDURE MakeMove(r,c:integer);
```

```
PROCEDURE TakeMove(var r,c:integer);
```

```
IMPLEMENTATION
```

```
VAR F:Array[1..3,1..3] of integer;
```

```
FUNCTION BeginGame:integer;
```

```
var r,c:integer;
```

```
begin
```

```

for r:=1 to 3 do
  for c:=1 to 3 do F[r,c]:=0;
randomize;
BeginGame:=random(2);
end;
PROCEDURE MakeMove(r,c:integer);
begin
  F[r,c]:=1;
end;
PROCEDURE TakeMove(var r,c:integer);
var rr,cc:integer;
begin
  for rr:=1 to 3 do
    for cc:=1 to 3 do
      if F[rr,cc]=0 then begin F[rr,cc]:=1; r:=rr; c:=cc; Exit end;
    end;
  end;
PROCEDURE EndGame(res:integer);
begin
end;
BEGIN
END.
//zeros.h
#include <stdlib.h>
int BeginGame();
void EndGame(int res);
void MakeMove(int r,int c);
void TakeMove(int &r, int &c);
int _F[3][3];
int BeginGame()
{

```

```

int r,c;
for(r=0;r<3;r++)
    for(c=0;c<3;c++) _F[r][c]=0;
randomize();
return random(2);
}

void EndGame(int res)
{
}

void MakeMove(int r,int c)
{
    _F[r-1][c-1]=1;
}

void TakeMove(int &r, int &c)
{
    for(r=0;r<3;r++)
        for(c=0;c<3;c++)
            if (_F[r][c]==0)
                {
                    _F[r][c]=1;
                    r++; c++;
                    return;
                }
}

```

Задача 11. «Гра на цукерки»

(Заочна олімпіада, вища ліга, 2005-2006)

Двоє дітей грають у цікаву гру на дошці розміру $M \times N$. ($2 \leq M \leq 50$, $2 \leq N \leq 50$). В кожній клітинці дошки знаходиться від 1 до 100 цукерок. Лише ліва верхня клітинка, яка має координати (1,1) - порожня.

Гра розпочинається з цієї клітинки.

Діти ходять по черзі. За один хід дозволяється переміститись на довільну кількість клітинок вправо або вниз і забрати собі усі цукерки, які знаходяться у клітинці, на яку відбулось переміщення; далі гра продовжується з цієї клітинки. Гра завершується, коли хтось дістанеться клітинки з координатами (M, N).

Перемагає той, хто збере більше цукерок.

Ігрове поле представлено у файлі **game.dat**: в першому рядку числа **M** і **N**; в наступних **M** рядках по **N** чисел - кількість цукерок у відповідній клітинці.

Ваша задача зіграти в цю гру за одного з дітей по вибору і виграти, якщо це можливо, або звести гру до нічиєї. Зрозуміло, що грати треба за того гравця, який має непрограшну стратегію.

Вказівки до розв'язку на мові Pascal.

При компіляції вашій програмі буде надано модуль **GAME.TPU**.

У цьому модулі буде реалізовано такі процедури:

```
PROCEDURE BeginGame(P : integer);
```

```
PROCEDURE MakeMove(R,C : integer);
```

```
PROCEDURE GetMove(var R,C : integer);
```

```
PROCEDURE EndGame;
```

Ваша програма повинна.

На початку гри викликати процедуру `BeginGame(p)`, де $p=1$, якщо ви хочете ходити першими, $p=2$ - другими.

Для того, щоб зробити свій хід, викликати процедуру `MakeMove(R,C)`, де параметри R та C - номер рядка та стовпчика, в який ви робите хід.

Для того, щоб дізнатись хід суперника викликати процедуру `GetMove(R,C)`, яка поверне у змінні R та C номер рядка та стовпчика, в який робить хід суперник.

Якщо гра закінчилась - викликати процедуру `EndGame`.

Ваша програма може виглядати таким чином:

```

USES Game;
...
BEGIN
...
BeginGame(K);
...
while...
begin
...
if ... then MakeMove(R,C)
else GetMove(R,C);
...
end;
...
EndGame;
END.

```

Вказівки до розв'язку на мові C++

При компіляції вашій програмі буде надано файл **GAME.H**

У цьому модулі буде реалізовано такі функції:

```

void BeginGame(int P);
void MakeMove(int R, int C);
void GetMove(int &R, int &C);
void EndGame();

```

Зміст цих функцій відповідає аналогічним процедурам з модуля GAME.TPU для роз'язків на мові Pascal.

Ваша програма може виглядати таким чином:

```

#include <game.h>
...
void main()
{

```

```

...
BeginGame(K);
...
while...
{
...
if ... MakeMove(R,C);
else GetMove(R,C);
...
}
EndGame();
}

```

Приклад.

game.dat

3 3

0 5 6

5 6 1

1 3 2

В даному випадку доцільно грати за першого гравця і робити перший хід в клітинку (1,3).

Задача 12. «Цікава гра»

(Заочна олімпіада, вища ліга, 2006-2007)

Нехай є деяке число N . За одну операцію кожному гравцю дозволяється ділити дане число націло (звичайне ділення з видаленням дробової частини результату) на будь-яке число, не менше за 2 і не більше за M . Програє той хто отримає в результаті 0. Гравці ходять по черзі. Знайдіть хто виграє, при оптимальній грі кожного з гравців.

Технічні вимоги:

Вхідний файл: **GAME.IN**

Вихідний файл: **GAME.OUT**

Обмеження часу: 1 секунда

Формат вхідних даних: у першому рядку вхідного файлу **GAME.IN** міститься єдине число K ($1 \leq K \leq 10000$) – кількість тестових блоків у файлі. В кожному наступному рядку міститься один тестовий блок, який складається з двох цілих чисел N та M , $2 \leq N \leq 1000000000$ та $2 \leq M \leq N$.

Формат вихідних даних: у файл **GAME.OUT** для кожного з тестових блоків у окремому рядку виведіть 1 , якщо при даних N та M виграє перший гравець, і 2 - якщо другий.

Приклад вхідних та вихідних даних:

GAME.IN	GAME.OUT
2	1
5 3	2
10 4	

Задача 13. «Податковий інспектор»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Приватний підприємець і податковий інспектор грають в наступну гру: Є матриця $N \times N$ в яку записані цілі числа P ($0 \leq P \leq 1000000$) - прибутки підприємця за певні періоди. Далі кожен гравець по черзі викреслює останній рядок, або останній стовпчик матриці, при умові, що сума чисел в цьому рядку/стовпчику парна. Учасник, який не зможе зробити хід (все вже викреслено, або сума в останньому рядку та останньому стовпчику непарна) програє гру. Напишіть програму, яка визначить хто виграє гру, при найоптимальнішій грі обох гравців.

Вхідні дані:

Перше число - T ($T \leq 10$)- кількість описів ігор. Далі йде T блоків. Кожен блок описує відповідну гру. Перше число блоку - N - розмір матриці. ($0 < N \leq 1000$), далі міститься $N * N$ чисел - матриця гри.

Вихідні дані:

У вихідний потік запишіть T чисел, по одному на кожен рядок. 1 - якщо виграє перший гравець, і 0 - якщо перший гравець програє.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
2	1
3	0
7 9 4	
1 3 3	
9 4 1	
3	
5 5 0 1 7	
1	
2	
2 5	

Теорія чисел

Задача 1.

(I обласна олімпіада, 1987 р.)

Задано три натуральних числа A , B і N . Скласти алгоритм знаходження натуральних чисел, які не перевищують N , які можна представити у вигляді суми (довільного числа) доданків, кожне з яких буде A або B . Якщо натуральне число M менше N можна представити різними способами, то його враховувати один раз.

Задача 2.

(I обласна олімпіада, 1987 р.)

Задано натуральне число N . Написати алгоритм, який перевіряє чи є дане число парним або непарним.

Задача 3.

(I обласна олімпіада, 1987 р.)

Дана цілочисельна таблиця $A[1:100]$. Написати алгоритм, який підраховує кількість таких елементів, для яких всі попередні елементи менші за $A[i]$.

Задача 4.

(I обласна олімпіада, 1987 р.)

Написати програму за допомогою якої можна знайти кількість натуральних чисел, менших натурального числа N , куб кожного з яких можна представити у вигляді суми квадратів трьох натуральних чисел (наприклад, $M = I + J + L$). Натуральне число, яке задовільняє кілька способів предствалення його куба, враховувати лише один раз

Задача 5.

(II обласна олімпіада, 1988 рік)

Написати алгоритм, який підраховує кількість всіх простих чисел, які не перевищують даного числа M .

Задача 6.

(II обласна олімпіада, 1988 рік)

Написати алгоритм визначення кількості натуральних чисел, які не перевищують M та задовольняють рівність $I - J = (K + L)$.

Задача 7.

(II обласна олімпіада, 1988 рік)

Серед цілих чисел $n, n+1, \dots, 2n$ знайти всі такі, які можна представити як суму квадратів двох цілих чисел.

Задача 8.

(II обласна олімпіада, 1988 рік)

Серед цілих чисел $n, n+1, \dots, 2n$, знайти всі прості числа, різниця між якими дорівнює двом.

Задача 9.

(II обласна олімпіада, 1988 рік)

Написати програму виведення всіх досконалих чисел від M до N , де $M < N$. Натуральне число називається досконалим, якщо воно дорівнює сумі своїх дільників, крім самого себе (Наприклад, $6 = 1+2+3$).

Задача 10.

(II обласна олімпіада, 1988 рік)

Знайти найменше ціле число A , яке для різних цілих чисел k, l, m, n задовольняє умові: A ділиться на k , $A+1$ ділиться на l , $A+2$ ділиться на m , $A+3$ ділиться на n .

Задача 11.

(II обласна олімпіада, 1988 рік)

Дано рівняння виду $m+(m+1)+(m+2)+\dots+(m+k)=a$. Скласти програму, що перевіряє, чи існує для даних натуральних k і a таке натуральне m , щоб k і m були рішенням рівняння. За допомогою програми знайти всі рішення рівняння при $a=1000$.

Задача 12.

(II обласна олімпіада, 1988 рік)

Серед цілих чисел $n, n+1, \dots, 2n$ знайти такі, які можна представити як суму квадратів двох цілих чисел. Вирішити завдання при $n=5, n=10$.

Задача 13.

(III обласна олімпіада, 1989 рік)

Написати програму виведення всіх тризначних десяткових чисел, сума цифр яких дорівнює даному цілому числу M .

Задача 14.

(IV обласна олімпіада, 1990 рік)

Розробити алгоритм, що дозволяє розкласти будь-яке натуральне число K ($K \geq 2$) на прості множники.

Результат розв'язання дозволяється представити одним із двох способів:

- 1) таблицею множників;
- 2) кожен отриманий множник виводити командою ДРУК (ім'я змінної)

Задача 15. «Кількість досконалих»

(V обласна олімпіада, 1991 рік)

Натуральне число називається досконалим, якщо воно дорівнює сумі всіх своїх дільників, включаючи 1. Скласти алгоритм знаходження всіх досконалих чисел, що не перевищують числа m .

Задача 16. «Прості множники»

(VII обласна олімпіада, 1993 рік)

Побудувати алгоритм розкладу натурального числа на прості множники.

Задача 17. «Число Армстронга»

(VII обласна олімпіада, 1993 рік)

Число Армстронга – таке число із k цифр, для якого сума k -х степенів його цифр дорівнює самому числу.

Наприклад: $153 = 1^3 + 5^3 + 3^3$

Знайти всі числа Армстронга з двох, трьох і чотирьох цифр.

Задача 18. «Бики і корови»

(VII обласна олімпіада, 1993 рік)

Скласти програму гри «Бики і корови». ЕОМ придумує чотиризначне число, в якому всі цифри різні. Гравець повинен відгадати це число, роблячи кілька спроб (вводячи числа в ЕОМ). Після введення чергового числа ЕОМ робить аналіз і повідомляє про ступінь співпадання введеного числа з початковим, тобто числа «биків» і «коров». «Корова» – це цифра в числі гравця, що співпала по розряду з такою ж цифрою у відгадуваному числі. «Бик» – цифра в числі гравця, що не співпадає по розряду з такою ж цифрою у відгадуваному числі. Якщо, наприклад, задане число 6482, то число 5428 містить 1 «корову» (цифру 4) і 2 «бики» (цифри 8 і 2).

Задача 19. «Досконале число»

(VII обласна олімпіада, 1993 рік)

Натуральне число називається досконалим, якщо воно дорівнює сумі всіх своїх дільників, не рахуючи його самого (наприклад, $6 = 1 + 2 + 3$ – досконале число). Запишіть алгоритм, що перевіряє, чи є дане число досконалим.

Задача 20. «Системи числення»

(VII обласна олімпіада, 1993 рік)

Запишіть алгоритм переведення числа m з основою системи числення p в число з основою системи числення q .

Задача 21. «Множина»

(XI обласна олімпіада, 1997 рік)

Нехай x – деяке натуральне число, а Mx – множина чисел, що утворюються з числа x перестановкою його цифр. Будемо називати число y безпосередньо наступним для x , якщо:

1. y належить Mx ;

не існує числа y_1 , що належить Mx і для якого виконується співвідношення

$$x < y_1 < y.$$

Будемо називати число z безпосередньо попереднім для x , якщо:

1. z належить Mx ;

не існує числа z_1 , що належить Mx і для якого виконується співвідношення

$$z < z_1 < x.$$

Зауваження. Числа розглядаються в десятковій системі числення. Нулями ліворуч від першої ненульової цифри нехтуємо.

Дано: натуральне x .

Отримати: y , що є безпосередньо наступним для x , z , що є безпосередньо попереднім для x .

Задача 22. «НСК»

(XIII обласна олімпіада, 1999 рік)

Задані два натуральні числа A і B ($1 \leq A \leq B \leq 10000$). Потрібно обчислити найменше спільне кратне (НСК) всіх натуральних чисел від A до

В включно, тобто знайти мінімальне натуральне число, яке ділиться на A , $A+1, \dots, B$.

Задача 23. «І знову Фібоначчі»

(XVII обласна олімпіада, 2003 рік)

Перше та друге числа Фібоначчі дорівнюють 1. Кожне наступне - це є сума двох попередніх. Отже, початок ряду Фібоначчі виглядає таким чином:

1 1 2 3 5 8 13 21 34 55 89 144 ...

Необхідно побудувати алгоритм і реалізувати його у вигляді програми FIB.PAS (BAS чи CPP), що знаходитиме найбільший спільний дільник N -го та M -го чисел Фібоначчі ($1 \leq N, M \leq 1000$).

У файлі FIB.DAT записано лише два числа N і M .

У файл FIB.SOL потрібно вивести шуканий спільний дільник.

Приклади:

FIB.DAT

12 18

7 10

20 10

FIB.SOL

8

1

55

Задача 24. «Найменше число»

(XIX обласна олімпіада, 2005 рік)

Записати найменше число, сума цифр якого дорівнює заданому числу N ($0 < N < 100000$). Вхідні дані: з стандартного вхідного потоку вводиться єдине число N .

Вихідні дані: вивести у стандартний вихідний потік найменше число, сума цифр якого дорівнює N.

Приклад 1:

Вхідний потік

6

Вихідний потік

6

Приклад 2:

Вхідний потік

10

Вихідний потік

19

Задача 25.

(Заочна олімпіада, 2000-2001)

Підрахуйте кількість останніх нулів в факторіалі числа N.

Вхідні дані:

число N ($0 \leq N \leq 2000000000$)

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Вихідні дані:

кількість останніх нулів.

Приклад вхідних даних:

8

Приклад вихідних даних:

1

Задача 26. Системи числення

(Заочна олімпіада, 2001-2002)

Написати програму для переведення числа з непозиційної римської системи числення у позиційну двійкову систему числення. Перевірити коректність вхідної інформації.

Вхідні дані: у файлі **INPUT.DAT** знаходиться послідовність латинських букв (не довша 255 символів), що може бути записом числа в римській системі числення.

Вихідні дані: у файл **OUTPUT.DAT** вивести послідовність знаків 0 і 1, які є записом даного числа у двійковій системі числення або запис *not correct record*, якщо вхідна інформація некоректна.

<i>Приклад 1.</i> INPUT.DAT XXIV OUTPUT.DAT 11000	<i>Приклад 2.</i> INPUT.DAT XXIIV OUTPUT.DAT not correct record
---	---

Задача 27 «Симпатичні простачки»

(Заочна олімпіада, 2002-2003)

Назвемо "симпатичними" числа у записі яких містяться лише непарні цифри. Простим називається число, яке має лише два дільники: 1 і саме число. Напишіть програму, яка друкує всі прості "симпатичні" числа із заданого інтервалу [M; N].

Технічні умови.

У єдиному рядку вхідного файлу NICE.DAT містяться два числа M і N розділені пропуском.

$$1 < M, N < \text{MaxLongint.}$$

У вихідний файл NICE.SOL записати знайдені числа. Кожне наступне число у новому рядку.

Задача 28. «Найменше»

(Заочна олімпіада, 2002-2003)

Дано N натуральних чисел, кожне з яких не перевищує 1000000000. $1 \leq N \leq 10000$. Знайдіть найменше натуральне число, яке не можна отримати у вигляді суми будь-яких з даних чисел.

Вхідні дані:

Вхідний потік в першій стрічці містить число N.

В другій стрічці вхідного потоку міститься N даних чисел розділених одним або більше пропусками.

Вихідні дані:

В першу стрічку вихідного потоку слід вивести одне число, яке вимагається в умові задачі.

Приклад даних:

Вхідні дані	Вихідні дані
5	7
3 8 1 11 2	

Задача 29. «Супер простачки»

(Заочна олімпіада, 2003-2004)

Візьмемо натуральне число S. Якщо воно просте (має два дільники: 1 і саме число), то шукаємо суму цифр цього числа. Якщо отримана сума є простим числом, то знову знаходимо суму цифр уже цього числа і т.д. Якщо цей процес не зупиняється, то таке число S назвемо "супер простим".

Наприклад "супер" простим є число 47, бо $47 \rightarrow 11 \rightarrow 2 \rightarrow 2 \dots$

Серед введеного набору натуральних чисел із інтервалу [N,M] знайти "супер" прості числа.

Технічні умови:

Файл-розв'язок не повинен перевищувати за розмірами 3 Кб.

Вхідні дані знаходяться в файлі "SIMPLE.DAT", в єдиному рядку якого записано два числа N і M ($1 < N, M \leq 2000000$), розділених пропуском.

Результат вивести у файл "SIMPLE.SOL", в кожному рядку якого записано "супер" просте число.

Приклад 1:

SIMPLE.DAT	SIMPLE.SOL
2 10	2
	3

5

7

Приклад 2:

SIMPLE.DAT	SIMPLE.SOL
2990 3120	2999
	3011

Задача 30. «В парі просто»

(Заочна олімпіада, 2003-2004)

Дано числа: 1, 2, 3, ... N ($N \leq 1000000$). Написати програму, що розподілить дані числа на

максимально можливу кількість пар таким чином, щоб сума чисел в кожній з них була

простим числом.

Вхідні дані: число N.

Вихідні дані: в кожному рядку по два числа розділені пропуском.

Приклад вхідних даних:

7

1 6

7 4

5 2

Приклад вихідних даних:

Задача 31. «Гігантоманія»

(Заочна олімпіада, 2003-2004)

Розкладіть на прості множники число $n!$ (n -факторіал), $2 \leq n \leq 100000$.

Примітка: факторіалом числа n є число, що дорівнює добутку всіх цілих чисел від 1 до n включно. Наприклад $5! = 1 * 2 * 3 * 4 * 5$.

Вхідні дані:

Вхідний потік містить в першій стрічці єдине число - значення n .

Вихідні дані:

Виведіть в одну стрічку вихідного потоку степені простих множників у вигляді $=\langle \text{множник} \rangle^{\langle \text{ступінь} \rangle} * \langle \text{множник} \rangle^{\langle \text{ступінь} \rangle} * \dots$ Множники у вихідному потоці повинні слідувати в порядку зростання.

Приклад:

Вхідний потік: 10

Вихідний потік: $=2^8 * 3^4 * 5^2 * 7^1$

Примітка:

дозволений час на проходження тесту для задачі 3 - 5 сек.

Задача 32. «Суперчисло»

(Заочна олімпіада, перша ліга, 2005-2006)

На "Іграх гігантів" проводиться інтелектуальний конкурс "Суперчисло", у якому необхідно в десятковому записі натурального числа N поміняти місцями тільки дві цифри, щоб одержане число стало якомога більшим.

Допоможіть українській команді "Maski.ua" виграти цей конкурс.

Вхідні дані: У текстовому файлі **supernum.dat** знаходиться лише одне число N . ($0 < N < 10^60000$).

Вихідні дані: У текстовий файл **supernum.sol** вивести одне число, отримане при перестановці цифр числа N .

Приклад вхідних і вихідних файлів:

Приклад 1.

supernum.dat	supernum.sol
2015	5012

Приклад 2.

supernum.dat	supernum.sol
56556	66555

Час на тест: 1с

Примітка: уважно слідкуйте за виведенням числа у файл. Перевірка буде здійснюватись побайтово, тому у вихідному файлі не повинно бути зайвих символів.

Задача 33. «Терези»

(Заочна олімпіада, вища ліга, 2005-2006)

Є терези і набір гир масою 1, 3, 9, 27, ..., $3^{(n-1)}$ кг, причому кожна гиря в єдиному екземплярі. На лівий важіль терезів кладеться предмет масою m кг ($0 \leq m \leq 3^{(n-1)}$). Потрібно розподілити гирі на терезах так, щоб був досягнутий баланс. Не вимагається використовувати всі гирі.

Формат вхідних даних: у першому рядку файлу **Weight.in** - числа m і n ($n \leq 20$) (цілі невід'ємні), розділені символом "пробіл".

Формат вихідних даних: у першому рядку файлу **Weight.out** - число m і маси гир на лівому важелі в порядку зростання. У другому рядку - маси гир на правому важелі терезів у порядку зростання.

Приклад введення і виведення:

Weight.in	Weight.out
5 3	5 1 3 9

Задача 34. «Подільність»

(Заочна олімпіада, перша ліга, 2005-2006)

Одна царська особа, познайомившись із двійковою системою і отримавши інтелектуальну насолоду від таблички множення у цій системі (таке буває), вирішила перевірити ознаки подільності. Перша думка: «З якого числа розпочати перевірку». Оскільки царство святкувало 31-річчя особи, то придворний математик порадив розпочати саме з цього числа (наша порада була б іншою, чи не так?). Але... свято є свято.

Задано двійкове число. Перевірити його подільність на 31.

Вхідні дані: В єдиному рядку текстового файлу **bin.in** записане двійкове число.

Вихідні дані: У стандартний вихідний потік вивести «yes» у разі коли дане число ділиться на 31, або «no» в іншому випадку.

Приклад вхідних і вихідних даних:

Приклад 1	Приклад 2
bin.in	bin.in
1011101	111110111
yes	no

Задача 35. «Цифри»

(Заочна олімпіада, вища ліга, 2006-2007)

Для кожної цифри вказано число, скільки разів її необхідно використати.

Побудуйте три числа:

- 1) найменше, яке можна побудувати з цих цифр;
- 2) найбільше, яке можна побудувати з цих цифр;
- 3) число, найближче до середнього арифметичного двох перших отриманих чисел, яке можна побудувати з цих цифр.

В десяти рядках текстового файлу **CIPHERY.IN** знаходяться числові значення кількості цифр відповідно для 0, 1, ..., 8, 9. Кількості не перевищують 100.

В файл **CIPHERY.OUT** в трьох рядках виведіть три числа відповідно: найменше, найбільше, одне з найближчих до середнього арифметичного найменшого та найбільшого.

Приклад файлу CIPHERY.IN:

2
2
1

0
0
0
0
0
0
0

Приклад файлу CIPHRU.OUT:

112
21100
10210

Задача 36. «Поліндроми»

(Заочна олімпіада, перша ліга, 2006-2007)

Для економії пам'яті в новому комп'ютері числа замінюються на близькі за значенням числа-поліндроми, і як результат запам'ятовується лише перша половина отриманого поліндрома. Вам потрібно написати програму для описаного комп'ютера, яка швидко по заданому числу знаходить найменший поліндром, який більший або рівний цьому числу.

Вхідні дані. в першому рядку файлу **number.dat** записане єдине натуральне число, кількість цифр якого не перевищує 1000.

Вихідні дані. в перший рядок файлу **number.sol** виведіть шуканий поліндром.

Приклад вхідних і вихідних даних:

number.dat

1711

number.sol

1771

Задача 37. «Автоморфні числа»

(Заочна олімпіада, перша ліга, 2006-2007)

Складіть програму знаходження всіх автоморфних чисел на відрізку $[m, n]$. Автоморфним називається число, що дорівнює останнім цифрам свого квадрату. *Наприклад:* $5*5=25$, $6*6=36$, $25*25=625$.

У файлі **Z3.dat** містяться числа m і n ($1 \leq m, n \leq 40000$), розділені одним або декількома пропусками.

У файл **Z3.sol** запишіть через пропуск всі автоморфні числа, що належать вказаному відрізку.

Приклад вхідних і вихідних даних:

Z3.dat

3 50

Z3.sol

5 6 25

Задача 38. «Послідовність 01»

(Заочна олімпіада, вища ліга, 2006-2007)

Визначити скільки існує послідовностей з 0 та 1 довжиною n , в якій не зустрічається двох одиниць підряд. $n < 45$.

Вхідні дані. В першій стрічці вхідного потоку міститься єдине число n .

Вихідні дані. В першу стрічку вихідного потоку вивести єдине число -- кількість послідовностей довжини n , в яких не зустрічається двох одиниць підряд.

Приклад вхідних даних:

5

Приклад вихідних даних, що відповідають вхідним:

13

Задача 39. «Кролики»

(Заочна олімпіада, перша ліга, 2006-2007)

Є пара кроликів. Пара починає щомісяця народжувати нову пару на третій місяць життя. Одна пара кроликів живе k місяців. Скільки кроликів буде через n місяців?

Технічні умови. У стандартному вхідному потоці міститься два цілі числа k і n ($3 \leq k \leq n \leq 30$). У стандартний вихідний потік вивести єдине ціле число – кількість кроликів через n місяців.

Вхідні дані

6 6

Вихідні дані

14

Задача 40. «Множити чи не множити»

(Заочна олімпіада, вища ліга, 2006-2007)

Дослідіть, чи існують числа в системі числення з основою M такі, що закінчуються цифрою N , більші за N і мають наступну властивість: результат добутку числа на цифру N можна отримати відрізавши останню цифру і дописавши її на початку.

Технічні обмеження:

M – натуральне число від 2 до 36 включно;

N – одне з наступного:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z.

Вхідні данні: N та M в першому рядку текстового файлу **INPUT.TXT**

Вихідні данні: в перший рядок файлу **SOLUTION.DAT** записати найменше з існуючих натуральних чисел або NO, якщо числа не існують.

Приклад 1 файлу **INPUT.TXT**:

4 10

Приклад 1 файлу **SOLUTION.DAT**:

102564

Приклад 2 файлу INPUT.TXT:

0 8

Приклад 2 файлу SOLUTION.DAT:

NO

Задача 41. «Ключ-1»

(Заочна олімпіада, перша ліга, 2006-2007)

У зв'язку з тим, що фермери Дієтенко та Вампіров багато займалися обчисленнями, вони дуже любляли різні числові головоломки. Особливо любили обмінюватися шифрованими повідомленнями електронною поштою. Ключ до шифру потрібно було обчислити якнайшвидше, тому що повідомлення зберігалось у поштовій скриньці лише два дні. Одного разу фермер Дієтенко прислав фермеру Вампірову чергову шифровку та деяке ціле число N . Ключем до шифру було найменше додатне ціле число, добуток цифр якого дорівнює N . Допоможіть фермеру Вампірову прочитати повідомлення.

Вхідні дані: у текстовому файлі **Z1.dat** міститься число N ($0 \leq N \leq 2147483647$)

Вихідні дані: у текстовий файл **Z1.sol** записати шукане ціле число або 0, якщо такого числа немає.

Приклад вхідних та вихідних даних:

1)

Z1.dat	Z1.sol
0	10

2)

Z1.dat	Z1.sol
5	5

3)

Z1.dat	Z1.sol
--------	--------

21	37
----	----

4)

Z1.dat	Z1.sol
11	0

Задача 42. «Найкращий кролик»

(Заочна олімпіада, перша ліга, 2006-2007)

Як відомо, фермери Дієтенко і Вампіров постійно конкурували один з одним. Кожен хотів довести іншому, що його кролики кращі. Але, у зв'язку з тим, що обидва мали неабиякий хист до математики, просто шукати найбільшого кролика їм було нецікаво. Тому найкращим вважався той кролик, вага якого виражалася простим числом та при цьому сума цифр цього числа була якомога більшою. Для перевірки фермери вибрали по N своїх кроликів. Допоможіть фермерам Дієтенку і Вампірову розв'язати цю суперечку і вибрати найкращого кролика.

Вхідні дані: у першому рядку текстового файлу **Z3.dat** міститься число N - кількість кроликів, яких представив кожен фермер (значення N не перевищує 1000). Далі слідує $2*N$ рядків, у кожному з яких записана маса одного кролика. Для зручності фермери записували масу кролика у грамах. Тому всі числа – натуральні. Причому у перших N рядках записані дані про кроликів фермера Вампірова, а в наступних N рядках - кролики фермера Дієтенка.

Вихідні дані: у перший рядок текстового файлу **Z3.sol** запишіть букву **V**, якщо переміг фермер Вампіров, або букву **D**, якщо переміг фермер Дієтенко. У другий рядок запишіть масу кролика-переможця.

Якщо у жодного кролика маса не буде виражатися простим числом, або маси кроликів-переможців у фермерів співпадають, у файл **Z3.sol** запишіть єдиний рядок, що містить слово **"draw"**.

Приклад вхідних та вихідних даних:

1)

Z3.dat	Z3.sol
4	V
1123	9551
9551	
2252	
951	
5921	
1129	
1893	
2854	

2)

Z3.dat	Z3.sol
2	draw
2682	
1986	
8910	
1000	

Задача 43. «Числа-паліндроми»

(Заочна олімпіада, 2008-2009)

Назвемо число паліндромом, якщо його десятковий запис читається однаково зліва-направо і справа-наліво. Скільки з введених чисел є паліндромами?

Вхідні дані: в стандартному вхідному потоці записані числа в десятковій системі числення, кожне в новому рядку. Всі числа додатні і не перевищують 2 000 000 000.

Вихідні дані: в перший рядок стандартного вихідного потоку вивести шукану кількість чисел-паліндромів.

Приклад вхідних і вихідних даних:

Стандартний вхідний потік:

12

321

1

212

222

213

Стандартний вихідний потік:

3

Задача 44.

(Заочна олімпіада, 2008-2009)

Відомо, що число Сміта – це таке складене число, сума цифр якого дорівнює сумі цифр усіх його простих співмножників. Так, прикладом числа Сміта може служити 202, оскільки $2+0+2=4$, і $2+1+0+1=4$ ($202=2*101$).

Знайдіть кількість чисел Сміта на проміжку $[N,M]$, де $0 < N, M \leq 30000$.

Формат вхідних даних. У вхідному файлі input.txt містять числа N і M розділені пропуском.

Формат вихідних даних. У вихідний файл output.txt вивести кількість чисел Сміта.

Задача 45.

(Заочна олімпіада, 2008-2009)

Зіграємо в таку гру. Вам дається натуральне число N . Ви знаходите суму квадратів його цифр і отримуєте нове число. Потім з цим числом проробляємо ті ж самі операції. Яке число ви отримаєте після K таких кроків?

Формат вхідних даних. У вхідному файлі input.txt містяться натуральні числа N і K , що не перевищують 10^9 .

Формат вихідних даних. У вихідний файл output.txt вивести шукане число.

Приклад вхідних та вихідних даних.

input.txt

123 3

output.txt

50

Задача 46. «Факторіали»

(Заочна олімпіада, 2009-2010)

Факторіал числа N (позначається $N!$) визначається наступною рекурентною формулою:

При $N=0$, $N! = 1$

При $N>0$, $N! = (N-1)! * N$

Необхідно визначити найменше число факторіал якого закінчується рівно на K нулів.

Вхідні дані:

У вхідному файлі factorial.in дано ціле число K , ($0 \leq K \leq 500000000$)

Вихідні дані:

У вихідний файл factorial.out вивести найменше число, факторіал якого закінчується рівно на K нулів.

Приклад вхідних даних:

2

Приклад вихідних даних:

10

Задача 47. «Цікаві числа»

(Заочна олімпіада, 2009-2010)

Ім'я вхідного файлу:

interest.in

Ім'я вихідного файлу:

interest.out

Максимальний час роботи на одному тесті:

300 ms

Максимальний об'єм використаної пам'яті:

64 мегабайта

Максимальна оцінка за задачу:

20 балів

Існують натуральні числа, що закінчуються цифрою N , такі, що перенесення цифри N на початок числа приводить до збільшення числа в N разів.

Наприклад, число 102564, $N=4$, $410256=102564*4$.

Формат вхідних даних: у єдиному рядку вхідного файлу задано натуральне число N ($2 \leq N \leq 9$).

Формат вихідних даних: єдиний рядок вихідного файлу повинен містити одне число - найменше натуральне число, що задовольняє дану умову.

Приклад вхідних та вихідних даних:

interest.in

4

interest.out

102564

Задача 48. «Дивні числа»

(Заочна олімпіада, 2009-2010)

Ім'я вхідного файлу:

strange.in

Ім'я вихідного файлу:

strange.out

Максимальний час роботи на одному тесті:

600 ms

Максимальна оцінка за задачу:

30 балів

Вася називає число дивним, якщо кількість його дільників непарна, наприклад число 4 – дивне, воно має 3 дільники: 1,2,4. Задано список чисел, Вася хоче знайти кількість дивних чисел у ньому. Так як Вася не вміє програмувати, то це завдання було доручене Вам.

Формат вхідних даних: у першому рядку вхідного файлу заданий розмір списку – натуральне число N ($1 \leq N \leq 100$). У наступних N рядках записані натуральні числа без ведучих нулів, які не перевищують 10^{20} .

Формат вихідних даних: єдиний рядок вихідного файлу повинен містити одне число - кількість дивних чисел у заданому списку.

Задача 49. «Eight 2 Sixteen»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

В файлі eight.dat записано дробове число в системі числення з основою 8. Файл може містити до 10^6 знаків.

В sixteen.sol виведіть це саме число в системі числення з основою 16.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
76543210.0101	FAC688.041

Фізика

Задача 1.

(II обласна олімпіада, 1988 рік)

Коло складається з двох опорів R_1 і R_2 . Ці опори з'єднані послідовно, якщо $n = 1$ і паралельно, якщо $n = 2$. Визначити загальний опір кола. Вирішити завдання для $R_1 = 15 \text{ Ом}$ і $R_2 = 21 \text{ Ом}$ і $n = 1$, $n = 2$.

Задача 2. «Драгстери»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Драгстерами називають надпотужні гоночні машини, які змагаються на дистанції 402 м. Драгстери всю дистанцію рухаються з прискоренням (будемо вважати його сталим) і перетинають фінішну пряму з швидкостями 250-500 км/год.

В першому рядку файла `dragster.in` записано число N – кількість драгстерів, в наступних N рядках через пропуск назви драгстерів і показники їх прискорення.

Обчисліть швидкість в км/год кожного драгстера з точністю до третього знака після десяткової крапки і виведіть назву самого швидкого. Результати виведіть в окремих рядках файла `dragster.out`. Гарантується, що переможець буде єдиний.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
3	456.505
"Flash" 20.000002	457.872
"Mega turtle" 20.12	468.001
"NoBreaks" 21.02	"NoBreaks"

Різне

Задача 1.

(III обласна олімпіада, 1989 рік)

Дітям потрібно визначити ведучого в деякій грі. Для цього вони, ставши в коло і визначивши першого, починають вважати за допомогою "лічилки", що містить M слів. Той, на кому закінчується "лічилка", виходить з кола. Після цього "лічилка" повторюється, причому з першого учасника. Якщо після чергового кола "лічилки" перший учасник вибув, то рахунок починається з наступного за ним учасника. Скласти алгоритм, що дозволяє визначити порядковий номер того з N гравців ($N < M$), який вибуває останнім.

Задача 2.

(IV обласна олімпіада, 1990 рік)

Колись дуже давно в шаховому королівстві всім Фігури ходили так, як заманеться. Але потім Королю набридли ці "заворушення" і він видав Указ, за яким кожній Фігура наказувалося ходити певним чином. Щоб цей Указ суворо виконувався, Король наказав кожен хід всіх Фігур записувати (що зараз і роблять шахісти), щоб можна було виявити порушника і за непокору зняти з Дошки.

Як-то раз Король наказав перевірити запис ходів Коня і доповісти, чи правильно Кінь обрав свій маршрут. Але ніхто в шаховому королівстві перевірити ходи Коня не зміг - адже жодної з Фігур (по Королівському Указу) не дозволялося ходити так, як Коневі.

Чи можете Ви, знаючи запис ходів Коня (у вигляді списку координат деяких N полів шахової дошки розміром 8×8), розробити алгоритм, що дозволяє визначити, чи є цей запис дозволеним маршрутом для коня?

Задача 3. «Обмін значеннями»

(V обласна олімпіада, 1991 рік)

Змінним X і Y присвоєні відповідно значення a та b . Складіть алгоритм, після виконання якого змінна X набуде значення b , а змінна Y набуде значення a . При цьому не можна використовувати додаткову змінну.

Задача 4. «Нова історія одного міста»

(VIII обласна олімпіада, 1994 рік)

Новий градоначальник міста Глупова вирішив з метою поповнення бюджету та економії пального провести компанію боротьби з лівим ухилом та лівими рейсами. Для цього він заборонив водіям виконувати ліві повороти, встановивши штраф за кожен поворот наліво в розмірі 1 мільйон та наказавши встановити систему тотального стеження за автомобілями, яка слідкує за кожним автомобілем і заносить його координати в пам'ять комп'ютера на початку та в кінці руху, а також в ті моменти, коли він виконує будь-який поворот. Від важкого минулого Глупову лишилися вулиці, що можуть перетинатися під будь-якими кутами. Рух у зворотньому напрямку градоначальник не заборонив.

ЗАВДАННЯ: написати програму, що для заданої послідовності координат автомобіля обчислює штраф водія.

Задача 5. «Едемський сад»

(IX обласна олімпіада, 1995 рік)

Едемський сад складається з N фруктових дерев, розміщення яких задано координатами (X_i, Y_i) , а їх врожайності, відповідно, дорівнюють U_i , $i=1,2,\dots,N$. Садівник обгородив сад огорожею мінімальної довжини. Розробити програму, яка виводить на екран план Едемського саду, на якому ілюструється взаємне розміщення огорожі і дерев. При цьому:

1. Забезпечити можливість введення початкових даних як з клавіатури, так і з файлу **EDEM.GOD**, і відображати їх на дисплеї у вигляді плану

Едемського саду (врахувати, що перший запис файлу **EDEM.GOD** вміщує значення N , а в кожному з наступних N записів вміщуються по три числа – X_i, Y_i, U_i , де $1 \leq i \leq N$, $N \leq 20$; числа в кожному записі розділені пропусками.

2. Забезпечити можливість діалогу редагування початкових даних з синхронним відображенням результатів редагування на плані Едемського саду.

3. Обчислювати і виводити на дисплей врожайність всього саду.

4. Обчислювати і виводити на дисплей максимальну відстань між деревами саду.

5. Обчислювати і виводити на дисплей мінімальну відстань між сусідніми деревами саду.

6. Визначати кількість рогів в найкоротшій огорожі.

7. Обчислювати і виводити на дисплей периметр огорожі саду.

8. Обчислювати і виводити на дисплей площу обгородженого саду.

9. Автоматично наносити на план саду найкоротший маршрут, додержуючись якого, можна обійти всі дерева і повернутися до місця старту, обчислювати відстань за цим маршрутом.

10. Динамічно відображати на плані обхід Едемського саду садівником вздовж знайденого найкоротшого маршруту.

Задача 5.

(X обласна олімпіада, 1996 рік)

Дано дві календарні дати. Визначити, які це дні тижня, та обчислити кількість днів між цими датами. Не забудьте, що високосним вважається рік, номер якого кратний чотирьом та не кратний 100, або кратний 400. Наприклад, між 21.01.1996 і 22.01.1996 - 1 день, між 01.02.1996 і 01.03.1996 - 29 днів, між 01.03.1995 і 01.02.1995-28 днів.

Дані читати з клавіатури в форматі "ДД.ММ.РРРР", де ДД - день, ММ - місяць, РРРР - рік. Перевіряйте коректність введеної інформації.

Задача 6. «Кодування»

(XVII обласна олімпіада, 2003 рік)

Хакер Вася шифрує свої повідомлення друзям наступним чином. Для кожного байту файлу він міняє місцями біти з номерами 0-1, 2-3, 4-5, 6-7. Так байт, значення якого 87, перетвориться на байт зі значенням 171. У файлі CODE.DAT записано закодоване таким чином повідомлення. (Розмір файлу CODE.DAT не більше 1024 байт). У файл CODE.SOL необхідно записати розкодоване повідомлення. Ваша програма повинна мати назву CODE.PAS (BAS чи CPP).

Приклад:

CODE.DAT

ABCDEFGH

CODE.SOL

ВБГИКЙЛД

Примітка: CODE.SOL це файл з 8 байт, що мають значення:

130 129 131 136 138 137 139 132

Задача 7. «Японський календар»

(XVII обласна олімпіада, 2003 рік)

У давньояпонському календарі був прийнятий 60-літній цикл, що складався з 12-літніх підциклів. Підцикли позначалися назвами кольорів: зелений, червоний, жовтий, білий, чорний. Всередині кожного підциклу роки мали назви тварин: пацюка, корови, тигра, зайця, дракона, змії, коня, вівці, мавпи, курки, собаки і свині. 1984 рік - рік зеленого пацюка - був роком початку чергового циклу. Напишіть програму CALENDAR.PAS (BAS чи CPP), що забезпечить введення деякого року з файлу CALENDAR.DAT та виведення його назви за давньояпонським календарем у файл CALENDAR.SOL

CALENDAR.DAT

2003

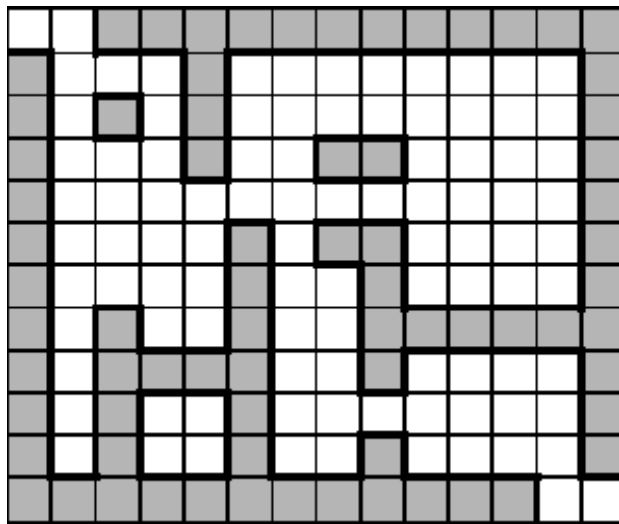
CALENDAR.SOL

Рік чорної вівиці

Примітка. При отриманні zatrudnень з виведенням у файл CALENDAR.SOL літер кирилиці допускається запис українських слів латинськими літерами. Наприклад, замість фрази "Рік зеленого пацюка" можна записати "Rik zelenogo pasyuka".

Зачачі 8. «Тяп_Ляп

(XX обласна олімпіада, 2006 рік)



Будівельна фірма «Тяп-Ляп» приймає замовлення на обробку стін внутрішніх приміщень (вирівнювання, штукатурка, фарбування і т.п.). Для розрахунку вартості робіт необхідно знати площу стін. Вам необхідно скласти програму, яка по заданому плану приміщення розраховує площу внутрішніх стін. План є прямокутною сіткою, на якій задано розташування стін. Ширина однієї клітини складає D метрів, висота стель скрізь однакова і дорівнює L метрів.

Вхідні дані:

В першому рядку вхідного потоку міститься два дійсні числа D , L . В другому рядку – два цілі числа N , M (довжина і ширина приміщення, задана в кількості клітин, $1 \leq N$, $M \leq 100$). В наступних N рядках містяться по M чисел, кожне з яких дорівнює 0 (клітина не належить стіні) або 1 (клітина належить стіні). Всі числа розділяються між собою пропуском.

Вихідні дані:

У вихідний потік необхідно вивести одне число – обчислену площу стін, з точністю до двох знаків після коми.

Наприклад (для схеми, наведеної в умові):

Input													Output
1													3
12													14
0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	1	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	1	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	0	0	1	0	0	0	0	1
1	0	1	0	0	1	0	0	0	0	0	0	0	1
1	0	1	0	0	1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0

Задача 9. «Банк»

(XX обласна олімпіада, 2006 рік)

Дідусь вирішив покласти в банк певну суму грошей під $k\%$ річних із таким заповітом: “Кожні 50 років знімати весь прибуток і витратити його на оновлення матеріально-технічної бази школи, яку я закінчив”.

Потім він вирішив підрахувати суму, що накопичиться після кожного року за умови, що сума вкладу 1000 грн. і $k = 5\%$. Три дні з ранку до вечора сидів дідусь за підрахунками, багато разів помилявся, починав обчислення спочатку, використав багато паперу, проте остаточно так і не підрахував. Тоді він звернувся до свого онука, який захоплювався програмуванням, брав участь у заочних олімпіадах і турнірах, успішно виступав у шкільних і районних олімпіадах і навіть отримав право захищати свій район на обласній олімпіаді з інформатики. Уявіть, що ви є тим онуком чи онучкою і допоможіть дідусеві справитися із цією задачею.

Формат вхідних даних.

У першому рядку стандартного вхідного потоку записане натуральне число S – сума грошей, яку дідусь поклав у банк ($1000 \leq S \leq 10000000$). У другому рядку записане число k ($0 < k < 1$) – відсоткова ставка.

Формат вихідних даних.

У єдиний рядок стандартного вихідного потоку треба вивести з точністю двох десяткових знаків прибуток із вкладу дідуся, який, відповідно до заповіту, отримуватиме школа через кожних 50 років.

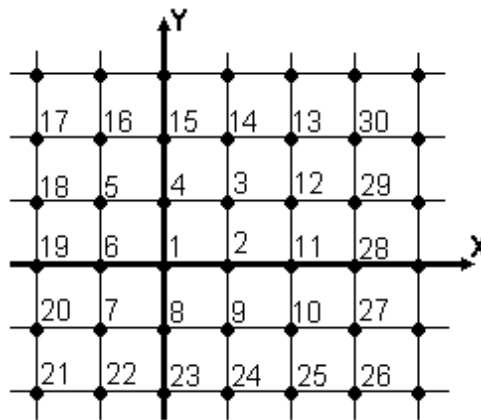
Приклад вхідних і вихідних даних:

Приклад 1	Приклад 2
Вхідні дані 1000 0.05	Вхідні дані 5000 0.10
Вихідні дані 10467.40	Вихідні дані 581954.26

Задача 10. «Нумерація координатної сітки»

(XXII обласна олімпіада, 2008 рік)

Вузли координатної сітки пронумеровані по спіралі, як показано на малюнку. Необхідно по координаті вузла визначити його номер.



Вхідні дані.

Ваша програма повинна вводити з клавіатури два цілих числа X і Y , які розділені пропуском, - координати вузлів по осі X і Y , відповідно ($-20000 < X < 20000$, $-20000 < Y < 20000$).

Вихідні дані.

Ваша програма повинна вивести на екран номер вузла, координати якого дорівнюють (X, Y) .

Приклад.

Задача 11. Доріжки*(Заочна олімпіада, 2001-2002)*

Одного разу в місті Хмельницькому випав великий сніг. І на ранок вся центральна площа була засипана снігом. Та зранку люди протоптали доріжки. Для того, щоб протоптати доріжку, по ній повинно пройти не менше 10 чоловік. Вам дана карта доріжок на площі. Визначіть найменшу кількість людей, що могли протоптати ці доріжки, якщо кожен з них проходив площею лише один раз, і не проходив по одному й тому ж сегменту доріжки більше одного разу.

Розміри площі -- 10 на 10 клітин.

Карта доріжок задається таким чином:

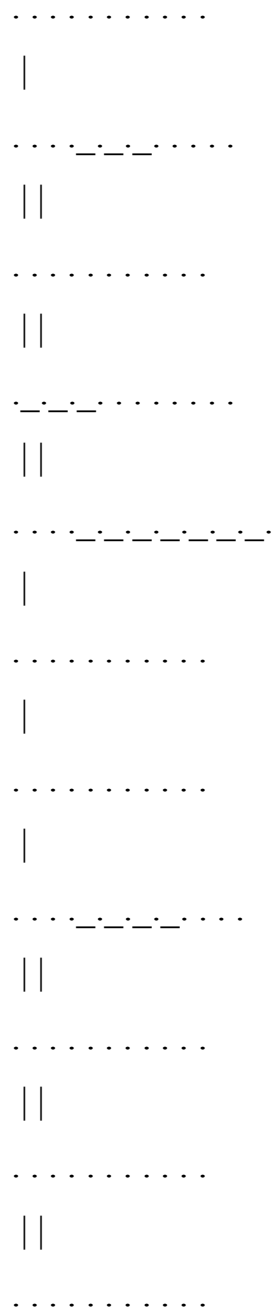
В першій стрічці вхідного потоку міститься 21 символ -- 11 крапок розділених рівно одним пропуском. В другій стрічці містяться символи '|', що являють собою вертикальні сегменти доріжок. Ці символи розміщені напроти деяких крапок першої стрічки, і являють собою сегмент доріжки, що з'єднує крапку першої стрічки з відповідною крапкою наступної стрічки. Всі інші символи другої стрічки -- пропуски. Третя стрічка знову складається з 21 символу -- 11 крапок розділених одним пропуском або знаком '_' (підкреслення), що представляє собою горизонтальний сегмент доріжки, який з'єднує відповідні крапки цієї стрічки. І т.д. Всього 21 стрічка у вхідному потоці. Перша і остання стрічки не можуть містити горизонтальних сегментів доріжок. Перший і останній стовпчик не може містити вертикальних сегментів доріжок (символ '|' не може бути першим та двадцять першим в стрічці). Іншими словами доріжка ніколи не проходить вздовж краю площі. Кожна людина зайшла на площу на одну з крайніх точок, переходила від однієї крапки до іншої, які безпосередньо з'єднані сегментами доріжок, не

пройшла більше одного разу ні по якому сегменту, та залишила площу з однієї з крайніх точок.

Вивести у вихідний потік одне число -- найменшу кількість людей, які могли протоптати задані доріжки.

Приклад.

Вхід:



Вихід: 25

Задача 12.

(Заочна олімпіада, 2001-2002)

В секретній лабораторії вивчають нову шкідливу сполуку. Зразки сполуки знаходяться в N пронумерованих контейнерах (в кожному контейнері один зразок). Деякі зразки вилучаються з досліджу, а нові добавляються – тому зразки сполук мають окрему нумерацію.

З цієї лабораторії в новий корпус передали зразок № K , але лаборант забув з якого контейнера була вийнята сполука. Напишіть програму для робота, за допомогою якої можна визначити номер порожнього контейнера.

Відомо:

- контейнери є всі: від 1 до N ($N < 1000$), - але стоять вони без порядку;
- в контейнерах зберігалися зразки з номерами від 1 до M ($M < 30000$), де $M > N$;
- для довільних двох контейнерів завжди виконувалася умова: в контейнері з більшим номером зберігався зразок з більшим номером.

Програмістам, що пишуть на мові Pascal буде надано модуль ROBOT.TPU, в якому буде реалізована така функція: `FUNCTION GetZrazokNumber(NumberOfKonteqner:Integer):Integer`

Програмістам, що пишуть на мові C++ буде надана бібліотека ROBOT.H, в якій буде реалізована така функція: `int GetZrazokNumber(int NumberOfKonteqner)`

Функція повертає номер зразка, що зберігається в контейнері з номером `NumberOfKonteqner`. Якщо повернений номер дорівнює 0, тоді пустий контейнер знайдено.

Данні K та N вводяться з клавіатури в одному рядку.

Номер пустого контейнера необхідно вивести на екран.

УВАГА! Викликати функцію можна певну обмежену кількість разів.

Тому наведений нижче приклад правильний, але не дає найкращого результату:

```

USES ROBOT;
VAR i, K, N : Integer;
BEGIN
  Readln(K,N)
  For i := 1 to N do If GetZrazokNumber(i)=0 Then Break;
  Writeln(i)
END.

```

В даній програмі оцінюється не тільки час виконання програми, а і кількість звернень до функції GetZrazokNumber. Максимальний бал отримає учасник, кількість спроб якого найменша. Інші пропорційно: **(бали за тест) * (мінімальна кількість) / (кількість учасника)**.

Задача 13.

(Заочна олімпіада, 2003-2004)

Пан Забувайко хотів потрапити в книгу рекордів і для того тренував свою пам'ять. На демонстрації феноменальної пам'яті випадкові глядачі протягом певного часу виголошували різні натуральні числа, а потім пан Забувайко повторював їх в довільному порядку.

Демонстрація провалилася – Забувайко забув одне єдине число.

Вам надається протокол демонстрації: послідовність з M натуральних чисел.

Відомо, що всі числа крім одного зустрічаються рівно 2 рази.

З'ясуйте, яке число не тримається в пам'яті пана Забувайко.

Вхідні дані: в першому рядку число M , в наступних – M натуральних чисел .

Вихідні дані: відповідне число.

Технічні обмеження: M і N не перевищують 2000000000.

Приклад вхідних даних:

9

1001

Приклад вихідних даних:

3000

5
3000
50
20000
50
20000
1001
5

Задача 14. «Змій»

(Заочна олімпіада, вища ліга, 2005-2006)

У країні Змієландії відбуваються щорічні змагання з проходження лабіринту зміями. Причому лабіринт є ламаною, ланки якої паралельні осям координат, і будь-які дві з них можуть мати не більше однієї спільної точки. Виграє змія найбільшої довжини, яка зможе подолати лабіринт. Причому очевидно, що деякі змії не зможуть продовжити свій шлях, так як на заваді їм буде стояти своє ж тіло. Перша і кінцева точки змії можуть одночасно знаходитися в одній точці. Довжина змії повинна не перевищувати довжину лабіринту.

Владислав захоплюється зміями і не міг не звернути увагу на ці змагання. За час до змагань завдяки своїм біологічним пізнанням хлопчик може виростити змія довільної довжини. Напишіть програму, яка визначить довжину змія, який потрібно виростити Владиславу, щоб гарантувати свій виграш в змаганнях.

Формат вхідних даних: у єдиному рядку вхідного файлу міститься число N ($0 \leq N \leq 4000$) - кількість пунктів лабіринту, через які він побудований. Кожні два сусідні пункти з'єднані прямою, яка паралельна осям координат. В наступних N рядках містяться по два числа x і y – координати пунктів лабіринту у порядку обходу, які за абсолютною величиною не перевищують 10000.

Формат вихідних даних: у єдиний рядок вихідного файлу виведіть єдине число – довжину змія з точністю до чотирьох знаків після коми.

Приклад введення і виведення:

snake.in	snake.out
5	4.0000
0 1	
3 1	
3 2	
2 2	
2 0	

Задача 15. «Центральна точка»

(Заочна олімпіада, вища ліга, 2006-2007)

На площині знаходиться n точок з цілими координатами (x_i, y_i) ; $i = 1..n$; $n \leq 1000$; $-10000 \leq x_i, y_i \leq 10000$. Назвемо центральною точкою одну з заданих точок з координатами (x_c, y_c) , таку, що для будь-якої іншої заданої точки (x_i, y_i) існує точка (x_j, y_j) , така, що $x_c - x_i = x_j - x_c$ та $y_c - y_i = y_j - y_c$ (тобто точки (x_i, y_i) та (x_j, y_j) є симетричними відносно точки (x_c, y_c)).

Задача. Визначити, чи існує серед заданих точок центральна точка.

Вхідні дані. В першій стрічці вхідного потоку міститься одне число n . В наступних стрічках містяться координати заданих точок, по одній точці в кожній стрічці. Координати точки є парою цілих чисел, що розділені одним або більше пропусками.

Вихідні дані. В першу стрічку вихідного потоку вивести координати центральної точки, розділивши їх одним або більше пропусками. Або вивести один знак "-", якщо у заданих точок не існує центральної точки.

Приклад вхідних даних:

6
0 4

2 6

1 3

2 2

2 6

0 0

Приклад вихідних даних, що відповідають вхідним:

1 3

Задача 16. «Перемикачі»

(Заочна олімпіада, вища ліга, 2006-2007)

Вибуховий пристрій, що був закладений невідомим терористом, має N - лампочок, та M - перемикачів. Кожний перемикач змінює стан деяких лампочок на протилежний. Для знешкодження бомби, необхідно вимкнути усі лампочки.

Напишіть програму, яка врятує світ.

Вхідні дані (файл *BOMBA.DAT*):

У першому рядку число N . ($0 < N \leq 100$).

У другому рядку міститься N чисел, що позначають стан кожної лампочки (0 - вимкнена, 1 - ввімкнена).

У третьому рядку міститься число M . ($0 < M \leq 100$).

В кожному з наступних M рядків міститься число K (кількість лампочок, стан яких змінює даний перемикач) і K чисел - номери відповідних лампочок.

Вихідні дані (файл *BOMBA.SOL*):

Якщо бомбу знешкодити неможливо, запишіть у вихідний файл "NO" (без лапок).

Якщо можливо врятувати світ, то у перший рядок запишіть "YES" (без лапок), а у другий рядок запишіть номери перемикачів які необхідно задіяти.

BOMBA.DAT	BOMBA.SOL
-----------	-----------

5	YES
0 1 1 0 1	1 3
4	
3 2 3 4	
3 1 4 5	
2 4 5	
2 2 4	

Задача 17. «Кролики-2»

(Заочна олімпіада, перша ліга, 2006-2007)

Земляни знайшли планету, придатну для життя і відправили космічний корабель з одним кроликом щоб переконатися у придатності для життя клімату планети. Кролику сподобався клімат і уже через місяць він привів ще одного (на цій планеті для цього достатньо одного кролика). Далі кролики почали розмножуватися з такою ж швидкістю – кожен кролик через місяць приводив ще одного. Але, розмноження кроликів почав контролювати монстр – корінний житель цієї планети. Як тільки на початку якогось місяця кроликів ставало більше ніж k , він поїдав k кроликів.

Земляни повернулися на планету через n місяців. Скільки кроликів вони нарахували?

Технічні умови. У стандартному вхідному потоці міститься два цілі числа n і k ($0 \leq n \leq 20$, $1 \leq k \leq 1000$), у стандартний вихідний потік вивести єдине ціле число – кількість кроликів через n місяців.

Вхідні дані

5 10

Вихідні дані

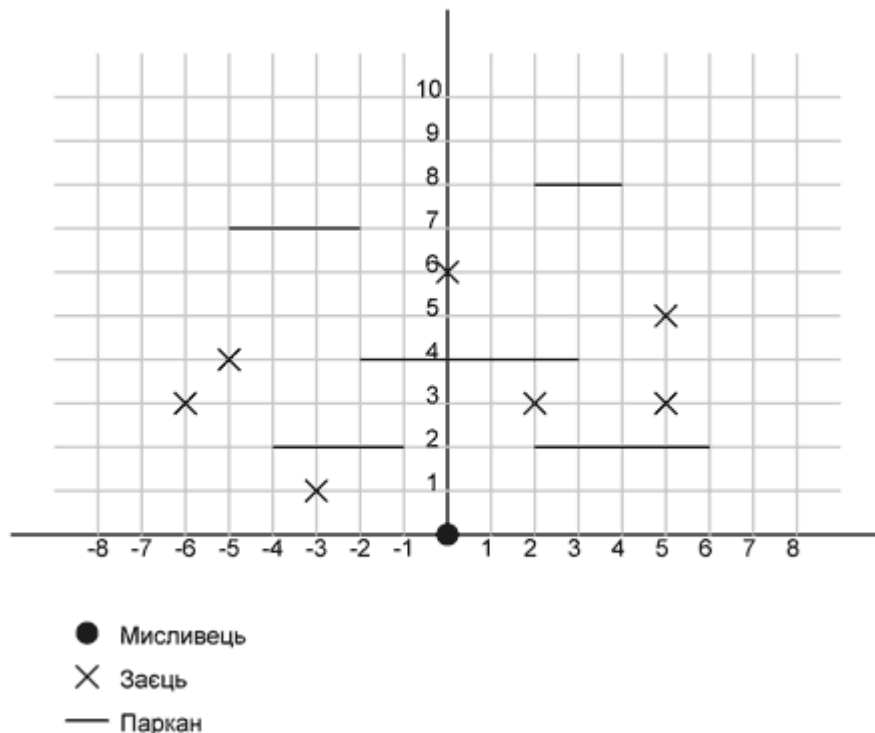
12

Пояснення. На кінець першого місяця кроликів стало 2, другого – 4... На початок 5 місяця їх було 16 і монстр з'їв 10. Отже, кроликів залишилося 6 і на кінець цього місяця добавилося ще 6. Маємо 12 через 5 місяців.

Задача 18. «Мільйон зайців»

(Заочна олімпіада, вища ліга, 2006-2007)

На абсолютно рівному полі в координатах $(0; 0)$ знаходиться мисливець. Поле перегороджене N парканами. Паркан починається в точці $(x_{pi}; y_{pi})$, $i = 1..N$, та продовжується на w_{pi} метрів паралельно осі X , в напрямку зростання координати x . Паркани є високими та непрозорими, так що мисливець не може бачити, що знаходиться за парканом. На полі за парканами ховається R зайців з координатами (x_{rj}, y_{rj}) , $j = 1..R$. Скількох зайців бачить мисливець?



У суперечливому випадку, коли $x_{rj} / y_{rj} = x_{pi} / y_{pi}$ або $x_{rj} / y_{rj} = (x_{pi} + w_{pi}) / y_{pi}$, вважати що зайця не видно, бо він добре заховався.

Початкові дані:

В першому рядку вхідного потоку міститься число N , $1 \leq N \leq 10000$. В наступних N рядках міститься по три цілих числа, розділених пропусками: x_{pi} , y_{pi} , w_{pi} . $-10000 \leq x_{pi} \leq 10000$, $0 < y_{pi} \leq 10000$, $w_{pi} > 0$, $-10000 \leq x_{pi} + w_{pi} \leq 10000$. В наступному рядку міститься число R , $1 \leq R \leq 10000$. В наступних R рядках міститься по два цілих числа x_{rj} , y_{rj} . $-10000 \leq x_{rj} \leq 10000$, $0 < y_{rj} \leq 10000$.

Результат:

У вихідний потік вивести одне число – скільки зайців бачить мисливець.

Приклад початкових даних:

Дані, що відповідають зображенню наведеному вище:

5
2 8 2
-5 7 3
-2 4 5
-4 2 3
2 2 4
7
0 6
5 5
-5 4
-6 3
-3 1
2 3
5 3

Приклад результату:

Результат, що відповідає даним наведеним вище:

2

Задача 19. «Дешифратор»

(Заочна олімпіада, вища ліга, 2006-2007)

До олімпійця звернувся його однокласник Вася (Вася не відвідував школу цілий місяць і йому вкрай необхідно отримати добру оцінку з контрольної), аби той вкрав комп'ютер вчителя і вкрав завдання контрольної роботи. Олімпієць Сашко (ім'я змінене, щоб його не взнав полковник Іваненко з наступної задачі), розуміючи, що це злочин, але міркуючи тим, що він неповнолітній, вирішив допомогти своєму другу Васі.

Сашко з першої спроби підібрав пароль до вчительського комп'ютера (ним виявилось слово 'password') і скопіював з нього коди доступу до контрольних робіт. Але як стало відомо, що з цих кодів лише два є вірними. Їх можна отримати один з одного циклічною перестановкою літер (коли деякі літери по черзі переносяться з кінця рядка в початок, наприклад: "abcdef" та "efabcd"). Вам необхідно визначити ці два ключа.

Вхідні дані: У першому рядку вхідного потоку число N ($3 \leq N \leq 10$). У кожному з наступних N рядків міститься код, довжина якого ненульова і не перевищує 10000.

Вихідні дані: у першій та другій рядки вихідного потоку запишіть вірні коди в тому порядку, в якому вони зустрілись у вхідному потоці.

Приклад вхідних даних:

```
3
efabcd
kukuku
abcdef
```

Приклад вихідних даних:

```
efabcd
abcdef
```

Задача 20. «Рукавиці»

(Заочна олімпіада, 2008-2009)

На швейній фабриці виготовляють рукавиці двох розмірів: для дорослих та для дітей. Над цим працюють 4 цехи: у першому шиють ліві рукавиці для дорослих, у другому - праві, у третьому - ліві рукавиці для дітей, а в четвертому - праві для дітей. Готова продукція по конвеєру потрапляє до великого ящика. По закінченні робочого дня робот-сортувальник аналізує вміст ящика і записує результати до текстового файлу. Виявивши велику ліву рукавицю, він дописує до файлу рядок з єдиною літерою L, велику праву - з літерою R, маленьку ліву - з літерою l (мала англійська L), маленьку праву - з літерою r.

Потрібно вдосконалити аналітичний модуль робота, написавши програму для визначення, скільки всього пар рукавиць пошито.

Вхідні дані: у кожному з рядків текстового файлу z2.txt записана одна з літер (L, R, l або r).

Вихідні дані: у єдиний рядок текстового файлу z2.sol вивести ціле число - загальну кількість пар рукавиць обох розмірів.

Задача 21. «Спекулянт»

(Заочна олімпіада, 2008-2009)

З дитинства нам розповідали (а вам, учням, і досі розповідають), що наш добробут залежить від того, наскільки ми будемо працювати.

Це була теорія, а практика показала, що наш добробут залежить не від інтенсивності нашої праці, а від курсу долара чи стрибків цін на нафту на світовому ринку.

Розглянемо валютного спекулянта на ім'я Баксік, який намагається заробити на економічній кризі.

В його країні є N міст в яких діють обмінні пункти валют.

Баксік аналітично прорахував які мають бути курси валют у цих містах на D днів вперед.

Він також має інформацію про ціни проїзду по кожній з доріг між містами (між кожними 2 містами є дорога).

За законами його країни, одна особа в день може здійснити лише одну валютну операцію. Тому баксік може здійснити обмін лише в одному місті країни, за один день.

На початку економічної кризи, баксік має UAH гривень, і USD доларів.

За день він може переїхати в будь-яке інше місто і здійснити там обмін валют, або лишитись у місті і здійснити обмін валют, або лишитись у місті і не здійснювати обміну валют.

Вам необхідно порахувати скільки максимально грошей у гривнях може назбирати Баксік через D днів.

Вхідні дані:

У першому рядку стандартного вхідного потоку записані числа N , D , UAH, USD. ($1 \leq D \leq 100$), ($1 \leq N \leq 100$) , ($0 \leq \text{UAH} \leq 10000$) , ($0 \leq \text{USD} \leq 10000$)

Наступні N рядків містять по N чисел і описують ціни проїзду по відповідним дорогам. (число в i -му рядку і j -му стовпчику показує ціну переїзду по прямій дорозі з міста i в місто j)

Кожен з наступних N рядків містить по D пар дійсних чисел - курси валют для відповідних міст на всі дні кризи. Перше число з кожної пари показує за скільки гривень куплять за 1 долар, друге число з пари показує - за скільки гривень продадуть 1 долар. Друге число в парі завжди не менше від першого.

Баксік на початку кризи знаходиться в місті з номером 1.

Вихідні дані:

У перший рядок запишіть максимальну суму в гривнях (округливши до двох знаків після коми), яку може назбирати Баксік в кінці кризи.

В кожному з наступних D рядків запишіть інформацію про діяльність Баксіка за кожний день в наступному форматі:

перше число рядка містить номер міста в яке перемістився (чи в якому лишився) Баксік.

друге число показує кількість доріг - R - яку він проїхав для цього переходу.

Наступні R чисел показують номери міс через які він проїжджав (в порядку проїзду, не зазначаючи першого міста).

Останні два числа показують скільки гривень та доларів він продав у цьому місті.

Приклад вхідних даних:

3 1 100 100

0 3 1

3 0 1

1 1 0

5.00 6.00

7.00 8.00

5.00 6.00

Приклад вихідних даних (увага, приклад може бути не оптимальним!!!):

798.00

2 2 3 2 0 100

Оцінювання: ваша програма принесе вам тим більше балів, чим більше грошей назбирає ваш Баксік :) в порівнянні з іншими...

Примітка: Обмін завжди здійснюється тільки до копійок (центів). Тобто якщо 1 долар продається за 3 гривні, то Баксік маючи 100 гривень може придбати лише 33.33 долари, заплативши при цьому 99.99 гривень, а одну копійку вже розміняти не можемо.

Задача 22. «Знову шифровка»

(Заочна олімпіада, 2008-2009)

Відомий метод шифрування квадратом: деяке повідомлення вписується у квадрат по рядках. При цьому пропуски замінюються знаком '*', а якщо в останньому рядку квадрата залишилися вільні клітинки, вони заповнюються крапками ('.'). Після цього, прочитавши вміст квадрата стовпець за стовпцем, отримують зашифрований текст.

Ваша задача полягає у тому, щоб зашифрувати вхідне повідомлення.

Вхідні дані:

У файлі Z1.dat міститься повідомлення, довжиною не більше 5000 символів. При зчитуванні символи переведення рядка необхідно ігнорувати.

Вихідні дані: У файл Z1.sol вивести зашифроване повідомлення.

Приклад:

Z1.dat:

Attack at dawn or we are done for

Проміжне подання у вигляді квадрата:

```
A t t a c k
* a t * d a
w n * o r *
w e * a r e
* d o n e *
f o r . . .
```

Задача 23.

(Заочна олімпіада, 2009-2010)

Заочна олімпіада з інформатики в Хмельницькій області проходить вже 10 років. Маємо надію, що вона буде готувати «інформатиків» ще гугол років.

Нехай відомо число N – кількість днів до певної заочної олімпіади з інформатики в далекому майбутньому. В який день тижня відбудеться ця олімпіада, якщо сьогодні четвер (15.10.2009).

Формат вхідних даних: в одному рядку – натуральне число N – кількість днів. ($N \leq 365 * 10^{100}$).

Формат вихідних даних: слово маленькими латинськими літерами – день тижня.

Приклад вхідних та вихідних даних:

Вхідні дані:

365

Вихідні дані:

friday

Список днів тижня англійською:

monday – понеділок

tuesday - вівторок

wednesday - середа

thursday - четвер

friday - п'ятниця

saturday - субота

sunday – неділя

Задача 24. «Cubes»

(Заочна олімпіада, 2009-2010)

У дитячому садку Вовочка любив складати із кубиків вежі. Кубики у нього були такі, що кожна грань мала свій оригінальний колір. Вовочка клав кубик на кубик так, щоб кожна грань його побудованої вежі була одного кольору. Зрозуміло, що кожного разу він міг скласти вежу різної висоти в залежності від того, які кольори для граней він вибирав. Цікаво, - подумав Вовочка, - а яку ж найбільшу вежу можна скласти з цих кубиків? Задача виявилася не дуже простою. Попробуйте скласти програму, що визначить максимальну висоту вежі, побудованої за описаними правилами, якщо ребро кубика рівне 1.

Формат вхідних даних. Перший рядок вхідного файлу `cubes.in` містить натуральне число N ($1 < N \leq 1000$) – кількість кубиків у наборі Вовочки. Далше у наступних N рядках описані кольори граней кожного кубика за допомогою шести великих літер латинського алфавіту, що позначають відповідний колір (A - Azure Blue, B - Blue, C - Cyan, G - Green, O - Orange, R - Red, S - Sun Yellow, V - Violet, W - White, Y - Yellow). Грані кубика ідуть у наступному порядку: передня, права, ліва, задня, верхня, нижня.

Формат вихідних даних. У вихідний файл `cubes.out` вивести найбільшу висоту вежі, що може бути побудована за описаними правилами з кубиків Вовочки.

Приклад вхідних та вихідних даних.

cubes.in

4

GYVABW

AOCGYV

CABVGO

OVYWGA

cubes.out

3

Задача 25. «Командна гра»

(Заочна олімпіада, 2010-2011)

Ліміт часу: 1 с

Ліміт пам'яті: 64 Мбт

Якось Петрик з однокласниками зібралися пограти у командну гру. Для цього їм потрібно було утворити декілька команд. Але учні почали сперечатись, хто до якої команди потрапить. Щоб розв'язати цю проблему, вирішили покластись на жеребкування. Кожен учень отримав набір різнокольорових карток. У команду потрапляють ті учні, у яких є картки однакового кольору. Петрика призначили капітаном однієї з команд. Він міг

запросити у свою команду всіх однокласників, у яких виявилися картки такого ж кольору, як і у нього. Вони, у свою чергу, могли запросити у команду однокласників, що мали картки такого ж кольору, як і у них, і так далі. Але, на жаль, деяким учням карток просто не вистачило... Звісно, Петрик, як капітан команди, бажав би бачити у команді якомога більше своїх друзів. Допоможіть Петрику з'ясувати, скільки чоловік буде у його команді?

Вхідні дані:

У вхідному файлі z3.dat задано: у першому рядку - кількість однокласників N ($1 \leq N \leq 100$), у наступних N рядках - опис кольорів карток, які є у однокласників. Всі кольори пронумеровані номерами від 1 до 50. Для кожного учня записано: спочатку число M_i ($0 \leq M_i \leq 50$), що визначає кількість карток, що має i -й учень, а після цього перераховують номери кольорів карток. Сам Петрик має номер 1.

Вихідні дані:

У вихідний файл z3.sol вивести одне число - кількість учнів у команді Петрика.

Приклад вхідних та вихідних даних.

Приклад вхідних даних:	Приклад вихідних даних:
6 3 2 7 11 2 1 5 2 5 8 3 7 5 10 0 2 19 25	4