

Чернігівський обласний інститут післядипломної
педагогічної освіти імені К. Д. Ушинського

Чернігівська обласна Інтернет-школа
«Юний програміст»

ЗБІРНИК ЗАДАЧ ТА РОЗВ'ЯЗАНЬ ІЗ ПРОГРАМУВАННЯ

ЧАСТИНА 1



choippo.cn.sch.in.ua

Чернігів – 2016

Автори:

Бондаренко С.М., Зуб В.В., Коваленко О.І., Ліпін В.П., Хорошок С.В., Хрол Н.П.

Рецензенти:

Горошко Ю. В., завідувач кафедри інформатики та обчислювальної техніки Чернігівського національного педагогічного університету імені Т. Г. Шевченка, доктор педагогічних наук, професор

Покришень Д. А., завідувач кафедри інформатики та інформаційно-комунікаційних технологій в освіті Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського, кандидат педагогічних наук, доцент

Загальна редакція:

Баранова О. Є., методист відділу інформатики, інформаційних технологій та дистанційного навчання Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського

Літош Ю. М., завідувач відділу інформатики, інформаційних технологій та дистанційного навчання Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського

Збірник задач та розв'язань із програмування / С.М. Бондаренко, В.В. Зуб, О.І. Коваленко та ін. ; за заг. ред. О. Є. Баранової, Ю. М. Літоша. – Чернігів: ЧОШПО імені К. Д. Ушинського, 2016. – Ч.1. –40 с.

*Рекомендовано до друку
вченою радою Чернігівського обласного інституту
післядипломної педагогічної освіти імені К. Д. Ушинського
(протокол № 1 від 25.02.2016 р.)*

ЗМІСТ

Вступ	4
Задачі з використанням методів побудови алгоритмів	
Зуб В. В., Бондаренко С. М.	
Проста задача?	5
Сир для Анфіси – 2	5
Квадрат і точки	6
Зоопарк	7
Задачі на базові структури мови програмування:	
розгалуження	
Коваленко О. І.	
Два кола	8
Нова шафа	9
Турист	10
Обмін	11
Рівень навчальних досягнень	12
Парні і непарні числа	12
Належність точки відрізка	13
Сума непарних чисел	14
Нова гра Сергійка	15
Перша чи остання?	16
Задачі на базові структури мови програмування: цикли	
Хорошок С. В.	
Збільшити на 2	17
Юний садівник	18
Юні програмісти	19
Бакси в банці	20
Змій Горинич	20
Постійна сума цифр	23
Перетворення	23
Гурток хорового співу	24
Цікавий добуток	25
Задачі з використанням підпрограм та масивів.	
Ліпін В. П.	
Цукерки	27
Клавіатура	28
Хто отримає «автомат»?	29
Змійка	31
Координати сусідів	32
Молоко та пиріжок	33
Бондаренко С. М.	
Мінер	34
Середній бал для фігуристів	35
Хрол Н. П.	
Максимум	37
Список використаних джерел	40

Вступ

Даний збірник містить задачі та розв'язання самостійних практичних робіт обласної Інтернет-школи «Юний програміст». Обласну Інтернет-школу започатковано як форму дистанційного навчання та підготовки учнів 7-11 класів до інтелектуальних змагань з інформатики та підвищення рівня знань із програмування. Самостійні практичні роботи учнів з розв'язування задач проводяться у формі змагання за допомогою системи автоматизованої перевірки виконаних завдань на сайті <http://www.e-olymp.com/> із підведенням рейтингу. Наприкінці вивчення кожної теми демонструються найоптимальніші розв'язання задач. Навчально - методичні матеріали Інтернет-школи розміщені на сайті <http://choippro.cn.sch.in.ua/>.

Під час практичних робіт відбувається розв'язування нестандартних задач, які потребують застосування різних алгоритмів програмування: сортування, динамічного програмування, довгої арифметики, задачі на графи, комбінаторики тощо. Зокрема, розв'язання задач з інформатики мають певні особливості: учень повинен спочатку побудувати її математичну модель, шляхом логічного та математичного умовиводів розробити алгоритм розв'язування задачі, реалізувати його певною мовою програмування. І розглядати кожну задачу слід не як привід для швидкого програмування, а як вправу для розмірковування, для тренування мислення, для набуття вмінь вибудовувати правильне й красиве рішення — це і є головним у навчанні школярів основ алгоритмізації та програмування [1]. Тому така форма занять Інтернет-школи створює ефективні педагогічні умови для підготовки учнів до змагальних випробувань.

У збірнику представлені алгоритми розв'язання задач, запропоновані тьюторами Інтернет-школи. Кожен алгоритм розв'язання вказаних задач є коректним.

Збірник розрахований на вчителів інформатики, слухачів курсів підвищення кваліфікації та учнів, які готуються до участі в інтелектуальних змаганнях з інформатики, а також додатково займаються програмуванням.

Задачі з використанням методів побудови алгоритмів

Ідеї розв'язань задач

Зуба В. В., учителя математики, директора Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста;

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста

ПРОСТА ЗАДАЧА?

Програма зчитує двоцифрове число і виводить через пропуск кожен цифру окремо.

Вхідні дані

Натуральне число на проміжку від 10 до 99 включно.

Вихідні дані

Спочатку першу цифру числа і через пропуск другу.

Ліміт часу **0.3** секунди

Ліміт використання пам'яті **256** МіВ

Вхідні дані	Вихідні дані
23	2 3

Розв'язання

Потрібно двоцифрове число розділити на цифри. Оскільки ми користуємося позиційною десятковою системою, то поділ числа на цифри виконується цілочисельним діленням даного числа на 10. Для цього використовуємо операції DIV та MOD.

```
var a:integer;  
begin  
  read(a);  
  writeln(a div 10,' ',a mod 10);  
end.
```

СИР ДЛЯ АНФІСИ – 2

Розрізаючи сир у задачі «Сир для Анфіси», у господаря залишалися куски сиру у вигляді прямокутного паралелепіпеда з різними цілими довжинами сторін. Готуючи нову страву із сиру для Анфіси, господарю прийшлося розрізати дані куски сиру на кубики зі стороною 1. Яку найменшу кількість розрізів приходилось йому робити, щоб розрізати дані куски сиру, якщо він кожного разу розрізав один кусок сиру на дві частини.

Вхідні дані

В єдиному рядку записано три числа **A, B, C** - довжини ребер куска сиру.
 $1 \leq A, B, C \leq 2000000000$.

Вихідні дані

Єдине число найменша кількість розрізів.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
2 3 4	23

Розв'язання

Задача з розділу лінійних програм, тому кількість розрізів не залежить від вибору порядку розрізання сторін і обчислюється за формулою. Виведемо її. Почнемо розрізати по стороні А і одразу розріжемо на максимальну кількість частин. Отримаємо А частин розміру $1 \cdot B \cdot C$ і виконаємо А-1 розрізів. Далі ріжемо кожен з отриманих кусків по стороні В. Отримаємо А*В кусків розмірами $1 \cdot 1 \cdot C$, виконавши при цьому всього А*(В-1) розрізів. Залишається виконати розрізання по стороні С. Буде виконано А*В*(С-1) розрізів. Усього маємо (А-1)+А*(В-1)+А*В*(С-1)=А-1+А*В-А+А*В*С-А*В=А*В*С-1 розрізів.

```
var a,b,c: int64;  
begin  
  read(a,b,c);  
  writeln(a*b*c-1);  
end.
```

КВАРТАЛ І ТОЧКА

Яку найбільшу кількість точок із цілочисельними координатами на аркуші в клітинку можна накрити квадратом зі стороною N клітинок?

Вхідні дані

Єдине число – сторона квадрату N ($1 \leq N \leq 10000$).

Вихідні дані

Максимальна кількість накритих клітин K.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
1	4

Розв'язання

Накреслимо на папері в клітинку квадратики зі сторонами 1, 2, 3, ... клітинок. У першому випадку ми закриємо 4 точки, у другому – 9, у третьому – 16... Як бачимо, ми отримуємо числа, що є точними квадратами: $2^2, 3^2, 4^2, \dots$ Тому для введеного числа А потрібно вивести квадрат наступного числа.

```
var a: int64;
```

```
begin
read(a);
writeln((a+1)*(a+1));
end.
```

ЗООПАРК

У зоопарку N кліток вишикувані в ряд. У зоопарку, крім інших тварин, живуть дві мавпочки, Слава та Юра. Слава та Юра завжди були великими друзями і сиділи в сусідніх клітках, але тепер вони посварились і більше не хочуть бачити один одного. Наглядач уже зібрався переселити їх у відповідності з їх бажанням, проте виникла проблема. Слава і Юра — дуже освічені мавпочки (кожна з них закінчила аж по вісім класів!), і вони неодмінно хочуть знати, скільки всього існує способів розселити їх так, щоб їх клітки не були сусідніми, і, звичайно, їх клітки повинні бути різними. Можна вважати, що всі N кліток доступні, інші мешканці зоопарку готові переїхати куди завгодно.

Наглядач спробував було порахувати сам, але збився з рахунку біля бегемотів. І цілком очевидно, що без вашої допомоги йому не обійтись!

Вхідні дані

У першому рядку вхідних даних міститься число N ($2 \leq N \leq 100$) — кількість кліток у зоопарку.

Вихідні дані

Виведіть одно число — кількість способів поселити Славу та Юру в різні клітки так, щоб ці клітки не були сусідніми.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
Sample 1	Sample 1
3	2
Sample 2	Sample 2
4	6

Розв'язання

Нехай Юра вибере першу клітку. Тоді Слава може вибрати будь-яку із решти $N-2$ не сусідніх кліток. Якщо ж Юра вибере другу – Славі залишається вибір із $N-3$ кліток і т.д. Маємо спадну послідовність послідовних натуральних чисел $N-2, N-3, N-4, \dots, 2, 1$. Знайдемо її суму за відомою формулою: $S=(N-2+1)/2*(n-2)=(N-2)*(N-1)/2$. Оскільки Юра та Слава можуть обмінятися місцями, то кількість варіантів зростає в 2 рази. Тобто, $S=(N-2)*(N-1)$.

```
var a: int64;
begin
```

```
read(a);
writeln((a-2)*(a-1));
end.
```

Задачі на базові структури мови програмування: розгалуження

Ідеї розв'язань задач

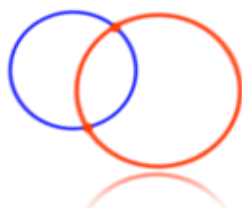
Коваленко О. І., учителя інформатики спеціалізованої загальноосвітньої школи № 2 з поглибленим вивченням іноземних мов м. Чернігова, учителя-методиста

ДВА КОЛА

Визначити в скількох точках перетинаються два кола.

Вхідні дані

6 чисел $x_1, y_1, r_1, x_2, y_2, r_2$, де x_1, y_1, x_2, y_2 - координати центрів кіл, r_1, r_2 - їх радіуси. Усі числа - дійсні, не перевищують 1000000000 за модулем, та задані не більш ніж із 3 знаками після коми.



Вихідні дані

Кількість точок перетину. Якщо точок перетину нескінченно багато, то вивести - 1.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

Вхідні дані	Вихідні дані
0 0 5 5 0 1	2

Розв'язання

Існує 4 випадки взаємного розташування кіл.

- 1) Обидва кола мають спільний центр і однаковий радіус.
- 2) Кола дотикаються одне до одного (відстань між центрами рівна сумі довжин радіусів).
- 3) Кола не дотикаються (або одне коло лежить повністю в середині іншого, або відстань між ними більша сумі радіусів).
- 4) В усіх інших випадках кола мають дві точки перетину. Так як ми передбачили всі інші випадки, тому нам не потрібно шукати спеціальних умов для даного випадку.

```
var x1,y1,r1,x2,y2,r2, d:real;
begin
  readln (x1,y1,r1,x2,y2,r2);
```



```

d:=sqr(x1-x2)+sqr(y1-y2);
if (x1=x2) and (y1=y2) and (r1=r2) then writeln (-1)
else
if (sqr(r1+r2)=d) or (sqr(r1-r2)=d) then writeln (1)
else
if (sqr(r1+r2)<d) or (sqr(r1-r2)>d) then writeln (0)
else
    writeln (2);
end.

```

НОВА ШАФА

Задано розміри прямокутних дверей **a, b** та розміри шафи, що має форму прямокутного паралелепіпеда **x, y, z**. Чи можна пронести шафу у двері, якщо проносити її дозволяється так, щоб кожне ребро шафи було паралельне або перпендикулярне кожній стороні дверей.

Вхідні дані

П'ять дійсних чисел **a, b, x, y, z** ($0 < a, b, x, y, z < 10$).

Вихідні дані

Вивести 1, якщо шафу можна вільно пронести у двері та 0 в протилежному випадку.

Ліміт часу 1 секунда

Ліміт використання пам'яті 64 МіВ

Вхідні дані	Вихідні дані
5 7 4 6 8	1

Розв'язання

Так як шафу можна проносити лише паралельно, стає очевидно, що існує лише три випадки які необхідно розглянути:

- 1) Чи можливо пронести шафу з параметрами (x;y);
- 2) Чи можливо пронести шафу з параметрами (z;y);
- 3) Чи можливо пронести шафу з параметрами (x;z).

program E76;

var a,b,c,x,y,z:real;

begin

 readln (a,b,x,y,z);

if ((x<a) and (y<b)) or ((x<b) and (y<a)) or ((z<a) and (y<b)) or
 ((x<a) and (z<b)) or ((y<a) and (z<b)) or ((z<a) and (x<b))

then writeln (1) **else** writeln (0);

end.

ТУРИСТ

Гена збирається на туристичний зліт учнів своєї школи. У своєму класі його було призначено відповідальним за палатки. У себе вдома він знайшов 3 палатки: перша з них важить a_1 кілограм і вміщує b_1 чоловік, друга важить a_2 кілограм і вміщує b_2 чоловік, третя важить a_3 кілограм і вміщує b_3 чоловік.

У класі Гени k чоловік. Виясніть, чи може він вибрати палатки так, щоб у них змогли поміститись усі. При цьому враховуйте, що вибрані палатки повинні разом важити не більше w кілограм.

Вхідні дані

Перший рядок містить два цілих числа k та w ($1 \leq k \leq 15$, $1 \leq w \leq 30$). Другий рядок містить шість цілих чисел:

$a_1, b_1, a_2, b_2, a_3, b_3$ ($1 \leq a_1, a_2, a_3 \leq 10$, $1 \leq b_1, b_2, b_3 \leq 15$).

Вихідні дані

Виведіть **YES**, якщо палатки вказаним способом вибрати можна, і **NO** у протилежному випадку.

Ліміт часу 1 секунда

Ліміт використання пам'яті 64 МіВ

Вхідні дані	Вихідні дані
10 10 5 5 6 6 4 5	YES

Розв'язання

Розглянемо всі можливі комбінації палаток. Якщо хоча б одна комбінація задовольняє умові, то Гена зможе помістити всіх своїх друзів.

program E206;

```
var k,w,a1,a2,a3,b1,b2,b3:integer;
    n1, n2, n3, n4,
    m1, m2, m3, m4 : longint;
begin
  read (k,w);
  read (a1,b1,a2,b2,a3,b3);
  m1 := a1 + a2; n1 := b1 + b2;
  m2 := a1 + a3; n2 := b1 + b3;
  m3 := a3 + a2; n3 := b3 + b2;
  m4 := a1 + a2 + a3; n4 := b1 + b2 + b3;
  if ( (a1<=w) and (b1>=k) ) or
     ( (a2<=w) and (b2>=k) ) or
     ( (a3<=w) and (b3>=k) ) or
     ( (m1<=w) and (n1>=k) ) or
     ( (m2<=w) and (n2>=k) ) or
     ( (m3<=w) and (n3>=k) ) or
     ( (m4<=w) and (n4>=k) )
```

```
then writeln('YES')
else writeln('NO');
```

end.

ОБМІН

У різдвяний вечір на підвіконні стояли три квіточки, зліва направо: герань, крокус та фіалка. Кожен ранок Маша витирала підвіконня і міняла місцями квіточку, що стояла праворуч, з центральною квіточкою. А Таня кожен вечір поливала квіточки і міняла місцями ліву та центральну квіточки. Потрібно визначити порядок квітів вночі після того, як пройде **k** днів.

Вхідні дані

Перший рядок містить кількість тестів **m** ($1 \leq m \leq 12$). У кожному з наступних **m** рядків знаходиться кількість днів **k** ($k \leq 1000$).

Вихідні дані

Вивести **m** рядків, що містять по три латинських літери: "G", "C" и "V" (великі літери без пропусків), які описують порядок квітів на вікні по закінченню **k** днів (зліва направо). Позначення: G – герань, C – крокус, V – фіалка.

Ліміт часу 1 секунда

Ліміт використання пам'яті 64 Мб

Вхідні дані	Вихідні дані
2	VGC
1	CVG
5	

Розв'язання

Необхідно побачити закономірність, що перестановки періодичні (період=3). Тому нас цікавить виключно три випадки, що і необхідно розглянути.

```
program E407;
```

```
var
```

```
  m, k, i: longint;
```

```
begin
```

```
  readln(m);
```

```
  for i := 1 to m do begin
```

```
    readln(k);
```

```
    if (k mod 3 = 1) then
```

```
      writeln('VGC');
```

```
    if (k mod 3 = 2) then
```

```
      writeln('CVG');
```

```

if (k mod 3 = 0) then
    writeln('GCV');
end;
end.

```

РІВЕНЬ НАВЧАЛЬНИХ ДОСЯГНЕНЬ

Установити рівень навчальних досягнень учня (початковий, середній, достатній, високий) відповідно до заданої оцінки (від **1** до **12**).

Вхідні дані

Одне число – бал учня.

Вихідні дані

Вивести **Initial** для початкового рівня, **Average** для середнього, **Sufficient** для достатнього і **High** для високого.

Ліміт часу **0.5** секунд

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
12	High

Розв'язання

```

program E902;

```

```

var x:integer;

```

```

begin

```

```

    readln (x);

```

```

    case x of

```

```

        1,2,3: writeln ('Initial');

```

```

        4,5,6: writeln ('Average');

```

```

        7,8,9: writeln ('Sufficient');

```

```

        10,11,12: writeln ('High');

```

```

    end;

```

```

end.

```

ПАРНІ ТА НЕПАРНІ ЧИСЛА

Задано три цілих числа **a**, **b**, **c**. Визначити, чи є серед них хоча б одне парне і хоча б одне непарне.

Вхідні дані

Числа **a**, **b**, **c**, які не перевищують за модулем **10000** (числа можуть бути від'ємними).

Вихідні дані

Вивести "YES" або "NO".

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
Sample 1 3 4 5	Sample 1 YES
Sample 2 7 7 7	Sample 2 NO

Розв'язання

```
program E1623;
```

```
var
```

```
  a, b, c: longint;
```

```
begin
```

```
  readln( a,b,c);
```

```
  if ((a mod 2 = 1) or (b mod 2 = 1) or (c mod 2 = 1)) and  
    ((a mod 2 = 0) or (b mod 2 = 0) or (c mod 2 = 0))
```

```
  then
```

```
    writeln('YES')
```

```
  else writeln('NO');
```

```
end.
```

НАЛЕЖНІСТЬ ТОЧКИ ВІДРІЗКУ

Визначити, чи належить задана точка відрізьку.

Вхідні дані

Шість цілих чисел – координати точки і координати початку і кінця відрізьку. Усі числа не перевищують за модулем **10000**.

Вихідні дані

Вивести "YES", якщо точка належить відрізьку, і "NO" у протилежному випадку.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
3 3 1 2 5 4	YES

Розв'язання

Якщо сума відстаней між точкою і кінцями відрізьків дорівнює довжині відрізьку, тоді точка належить, інакше ні.

Уточнення: При реалізації даної задачі на мові програмування C++, необхідно звернути увагу на встановлення умови:

if(d1+d2==d)...- неправильно

```
double eps = 1e-8;
```

if(d1+d2-d<eps) - правильно

```
program E2134;  
var x1,y1,x2,y2,x,y:real;  
begin  
  readln (x,y,x1,y1,x2,y2);  
  
  if      sqrt(sqr(x1-x)+sqr(y1-y))+sqrt(sqr(x2-x)+sqr(y2-y))=sqrt(sqr(x1-  
x2)+sqr(y1-y2))  
  then writeln ('YES') else writeln ('NO');  
end.
```

СУМА НЕПАРНИХ ЧИСЕЛ

Знайти суму непарних чисел на проміжку від **a** до **b**.

Вхідні дані

Два числа **a** та **b**, які за модулем не перевищують 10^9 .

Вихідні дані

Сума непарних чисел від **a** до **b**.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МБ

Вхідні дані	Вихідні дані
2 5	8

Розв'язання

Уважно придивившись, можна переформулювати задачу так:

Знайдіть суму арифметичної прогресії, де $\{a_1=A; a_n=B; d=2;\}$

АЛЕ! Якщо A дано парне, його необхідно збільшити на 1(привести до найменшого непарного, більшого A. Аналогічно з B (за необхідності зменшити).

Далі просто знайдемо суму арифметичної прогресії за формулою.

```
var  
  sum, A, B, count : int64;  
BEGIN  
  readln(A, B);  
  if A mod 2 = 0 then inc(A);  
  if B mod 2 = 0 then dec(B);  
  count := (abs(B - A) + 1);  
  if count mod 2 = 1 then inc(count);  
  count := count div 2;  
  if count mod 2 = 0 then  
  sum := (A + B) * (count div 2)
```

```

else
sum := (A + B) * (count div 2) + (A + B) div 2;
writeln(sum);
END.

```

НОВА ГРА СЕРГІЙКА

Трієчник Сергійко часто просить відмінника Васю зробити йому домашнє завдання. Так, як при усїй геніальності Васи він, усе ж таки, не може виконати роботу миттєво, то Сергійку приходитьсѧ чекати. Сергійку нудно чекати довго без справи. Не так давно він придумав нову гру, щоб не марнувати час.

На білий стіл, із прив'язаною до нього системою координат, Сергійко кладе прямокутний аркуш чорного паперу. Зверху на нього кладе прямокутний аркуш білого паперу, так що той можливо перекриває частину чорного аркуша. Сторони обох аркушів паперу паралельні осям координат. Після цього хлопчик вибирає точку на площині стола. Якщо вона попадає на чорний аркуш, Сергійко вважає, що очікування проходить не сумно, інакше він засмучуєтьсѧ. Допоможіть Сергійку зрозуміти, засмутитьсѧ він чи ні.

Вхідні дані

Спочатку з клавіатури вводятьсѧ координати лівого верхнього кута чорного прямокутника, потім правого нижнього, потім координати кутів білого прямокутника в тому ж форматі і наприкінці – координати точки. Усі координати – цілі числа, які по модулю не перевищують **10000**.

Вихідні дані

Програма повинна виводити слово "**SAD**", якщо Сергійко засмутитьсѧ (коли точка попадає на границю Сергійко вважає її такою, що належить чорному аркушу), та "**HAPPY**" – у протилежному випадку.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** Мів

Вхідні дані	Вихідні дані
2 10 5 3 4 4 6 1 3 8	HAPPY

Розв'язання

У даній задачі необхідно перевірити приналежність точки двом прямокутникам. Якщо точка належить лише одному (чорному), то Сергійко буде веселий, в усіх інших випадках він засмутитьсѧ.

Важливо звернути увагу на умову, та зрозуміти стан Сергійка у випадку якщо точка лежить на границі.

```

program E1657;
var x1,y1,x2,y2,x3,y3,x4,y4,x,y:longint;
begin
readln(x1,y1,x2,y2,x3,y3,x4,y4,x,y);

```

```

if (x3 < x) and (x < x4) and (y4 < y) and (y < y3) then
    writeln('SAD')
else
    if (x1 <= x) and (x <= x2) and (y2 <= y) and (y <= y1) then
        writeln ('HAPPY')
    else writeln('SAD');
end.

```

ПЕРША ЧИ ОСТАННЯ?

Задано трицифрове число. Визначити, яка цифра в ньому є більшою – перша чи остання.

Вхідні дані

У єдиному рядку задано трицифрове число.

Вихідні дані

Вивести більшу з вказаних цифр. У випадку їх рівності вивести знак "=" (без лапок).

Ліміт часу **0.5** секунд

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
328	8

Розв'язання

Існує багато різноманітних розв'язань даної задачі. Найпростіше знайти першу цифру за допомогою цілочисельного ділення на 100 ($\text{div } 100$), і останню за допомогою знаходження остачі при діленні на 10 ($\text{mod } 10$).

Далі необхідно просто їх порівняти.

```

program E903;
var x,y,z: integer;
begin
    read(x);
    y:=x div 100;
    z:=x mod 10;
    if y>z
    then writeln(y)
    else if y<z
        then writeln(z)
        else writeln('=');
end.

```


Задачі на базові структури мови програмування: цикли

Ідеї розв'язань задач

Хорошка С. В., учителя вищої категорії Коропської загальноосвітньої школи І-ІІІ ст. імені Т. Г. Шевченка

ЗБІЛЬШИТИ НА 2

Задано одновимірний масив **A** цілих чисел. Збільшити на **2** кожний невід'ємний елемент масиву.

Вхідні дані

У першому рядку задано натуральне число **h** - кількість елементів масиву ($h \leq 100$). У другому рядку через пропуск задано самі елементи масиву, значення кожного з яких за модулем не перевищує **100**.

Вихідні дані

В єдиному рядку вивести через пропуск **h** чисел: нові значення елементів масиву, у тому ж порядку, у якому їх було задано.

Вхідні дані	Вихідні дані
4 1 2 3 4	3 4 5 6

Пояснення: При виведенні результатів роботи числа повинні бути відокремлені одним пропуском, після останнього елементу пропуск повинен бути відсутнім.

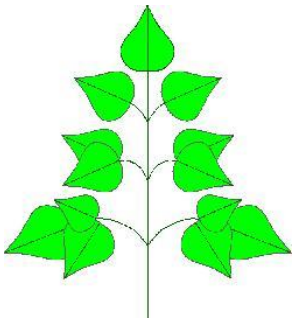
Розв'язання

Вводимо число. Використовуючи цикл **for**, вводимо елементи масиву. Порівнявши їх із нулем, виводимо додатні числа, збільшені на 2, від'ємні без зміни.

```
var a,i,h:integer;
begin
  readln(h);
  for i:=1 to h do
    begin
      read(a);
      if a>=0 then write(' ',a+2)
        else write (' ',a);
    end;
  end.
```

ЮНИЙ САДІВНИК

Мама попросила Васю полити всі молоді деревця в саду. Вася знає, що коли дерева маленькі, їх потрібно дуже добре поливати. А ось скільки поливати – невідомо. Але Вася – дуже розумний хлопчик. Він уважно прочитав весь підручник ботаніки для середньої школи і вияснив, що полив прямо пропорційний кількості листочків на дереві. Для гарного росту дерев достатньо вилити під дерево щоденно по одному літру води для кожного листка.



На щастя Васі виявилось, що листки на деревах ростуть ярусами, причому на верхньому ярусі два листка, на другому – чотири, на наступному - шість, і так далі, на кожному наступному ярусі на два листки більше в порівнянні з попереднім. А на самій верхівці росте ще один листочок. Хитрий Вася послав молодшу сестричку Марійку підрахувати кількість ярусів на кожному дереві, а Вас просить написати програму, яка для кожного дерева вирахає кількість літрів води для його поливу.

Вхідні дані

Кількість ярусів n ($0 \leq n \leq 1000$) на дереві.

Вихідні дані

Вивести кількість літрів води для поливу цього дерева.

Вхідні дані	Вихідні дані
3	13

Розв'язання

Для розв'язку цієї задачі потрібно знайти суму арифметичної прогресії, де $n=n$, $a_1=2$ і $d=2$, додавши до неї 1 (листок із верхівки). Для цього можна скористатися формулою суми арифметичної прогресії $S_n = \frac{2a_1+d(n-1)}{2}n \rightarrow n^2 + n$, або за допомогою циклу n раз додати номери ярусів, збільшені у два рази.

```
var i,n,s:int64;  
begin  
  readln(n); s:=1;  
  for i:=1 to n do  
    s:=s+i*2;  
  writeln(s)  
end.
```

ЮНІ ПРОГРАМІСТИ

Відомо, що у школі не менше ніж k_1 учнів, але не більше ніж k_2 учнів.



Також відомо, що кожен хлопчик дружить з n дівчатками, а кожна дівчинка з m хлопчиками. Яка мінімальна кількість учнів може бути у школі, і скільки у школі хлопчиків та дівчаток?

Юні програмісти, як Ви бачите, до цього часу розв'язують цю задачу. Допоможіть їм.

Вхідні дані

У першому рядку вхідного файлу знаходиться 4 числа, відокремлених

пропусками: k_1 , k_2 , n і m . Усі вхідні дані натуральні числа, які не перевищують 10000, $k_1 \leq k_2$.

Вихідні дані

У єдиному рядку вивести через пропуск три числа: спочатку кількість учнів у школі, потім кількість хлопчиків та дівчаток. Гарантується, що вхідні дані коректні і відповідь завжди існує.

Вхідні дані	Вихідні дані
20 30 4 5	27 15 12

Розв'язання

Збільшуємо кількість дівчаток (g) на 1, поки кількість хлопчиків (b) по формулі $b := g * m / n$ не стане цілим числом і сума хлопчиків та дівчаток буде більше або рівна мінімальній кількості учнів.

```
var k1, k2, n, g, m: int64;  
    b: real;  
begin  
  readln(k1, k2, n, m);  
  for g := n to k2 do  
    begin  
      b := g * m / n;  
      if (Frac(b) = 0) and ((b + g) >= k1) then break;  
    end;  
  writeln(b + g, ' ', b, ' ', g);  
end.
```

БАКСИ В БАНЦІ

Тато Карло подарував Буратіно 1 долар на його перший день народження, а заощадливий Буратіно поклав подарунок у банку. Кожного наступного року тато Карло подвоював свій попередній подарунок і додавав до нього стільки доларів, скільки років виповнилось Буратіно. А той, водночас, продовжував складати бакси в банку. На який N -й день народження в банці буде не менш, ніж S доларів?

Вхідні дані

Єдине число – значення S . $1 \leq S \leq 2^{40}$.

Вихідні дані

Шукане значення N .

Вхідні дані	Вихідні дані
15	3

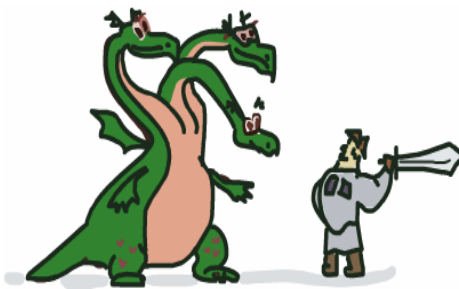
Розв'язання

У даній задачі виконуємо повторення, порівнюючи задану суму (S) із шуканою ($S1$) і збільшуючи її кожен раз на величину подарунку $p = p * 2 + n$, рахуємо рік (n), коли це станеться.

```
var s,s1,n,p:int64;  
begin  
  readln(s);  
  s1:=1;  
  p:=1;  
  n:=1;  
  while s1<s do  
    begin  
      inc(n);  
      p:=p*2+n;  
      s1:=s1+p;  
    end;  
  writeln(n);  
end.
```

ЗМІЙ ГОРИНИЧ

У далекому царстві жив Змій Горинич. У нього було N голів та M хвостів. Іван-Царевич вирішив знищити ворога людських душ. Для цього йому його кума Баба Яга подарувала чарівний меч, оскільки тільки ним можна вбити Змія Горинича. Якщо відрубати одну голову, то на її місці виростає нова, якщо відрубати хвіст, то



замість нього виросте 2 хвосту. Якщо відрубати два хвосту, то виросте 1 голова, і тільки коли зрубати 2 голови, то не виростає нічого. Змія Горинич гине тільки в тому випадку, коли йому відрубати всі голови і всі хвосту. Визначити мінімальну кількість ударів мечем, потрібну для знищення Змія Горинича.

Вхідні дані

Два числа N, M ($0 \leq N, M \leq 1000$).

Вихідні дані

Єдине число – мінімальна кількість ударів мечем, або -1, якщо знищити Змія Горинича неможливо.

Вхідні дані	Вихідні дані
3 3	9

Розв'язання

При рішенні даної задачі розглянуто декілька випадків:

- Змія Горинича неможливо вбити – це коли в нього непарна кількість голів і нема жодного хвоста. Якщо це так, то виводимо «-1», інакше переходимо до другого пункту рішення;
- Змія Горинича можна вбити – це коли в Змія є хоча б один хвіст або парна кількість голів. У такому випадку ми працюємо всередині циклу, при кожному проході якого перевіряється чи виконуються описані вище умови:

while (($N > 0$)**or**($M > 0$)) **do** {цикл буде виконуватися до тих пір, поки є хоча б одна голова чи хвіст}

Горинича можна вбити тільки тоді, коли при відрубуванні всіх хвостів залишиться парна кількість голів. Звідси, необхідно, щоб у Змія була парна кількість хвостів і сума голів і хвостів поділена на два, була парним числом.

Тому будемо працювати за даним алгоритмом:

1. Якщо кількість хвостів непарна, то відрубаємо один хвіст, на місці якого виросте два нових, і збільшуємо лічильник ударів на 1;
2. Якщо кількість голів непарна і хвостів більше одного, то відрубаємо два хвосту, тим самим збільшуємо кількість голів на 1 і збільшуємо лічильник ударів;
3. Якщо кількість голів і хвостів парна і якщо кількість хвостів при діленні на два дає парне число, то ми збільшуємо лічильник на $(M/2 + ((N + (M/2))/2))$, і прирівнюємо кількість хвостів і голів до нуля;
4. Якщо якась з умов пункту 3 не виконується, то збільшуємо кількість хвостів на 1, і збільшуємо лічильник на 1;
5. Повторюємо алгоритм до тих пір, поки не вб'ємо Горинича.

var N, M, count :integer;

begin

$\text{count} := 0$;

```

readln(N,M);
if ((N mod 2 <>0) and (M=0)) {перевірка чи можна вбити Горинича}
  then writeln(' -1') {вивести "-1"}
  else
    begin
      while ((N>0)or(M>0)) do {цикл буде виконуватися до тих пір, поки
є хоча б одна голова чи хвіст}
        begin
          if (M mod 2=1) {якщо число хвостів непарне, то відрубавши один
хвіст збільшуємо їх кількість на 1}
            then
              begin
                inc(count); {збільшення лічильника відрізів на 1}
                inc(M);
              end;
              if ((N mod 2=1) and(M>=2)) {якщо голів непарна кількість, то
відрубуємо два хвости(M) і збільшуємо кількість голів(N) на 1}
                then
                  begin
                    M:=M-2;
                    inc(N);
                    inc(count);
                  end;
                  if (M mod 2=0)and(N mod 2=0)and (((N+(M div 2))mod 2)=0 ) {якщо
число голів і хвостів непарне, і якщо кількість хвостів при діленні на два
дає парне число}
                    then begin
                      count:=count+(M div 2)+((N+(M div 2)) div 2); {то збільшуємо
лічильник надрізів}
                      N:=0;
                      M:=0; {і обнуляємо кількість голів і хвостів для виходу з
циклу}
                    end
                  else {якщо не виконується хоча б одна з умов минулого
розгалуження, то кількість хвостів збільшуємо на 1}
                    begin
                      inc(M);
                      inc(count);
                    end;
                  end;
                writeln(count); {виводим кількість ударів}
              end;
            end.

```

ПОСТІЙНА СУМА ЦИФР

Знайти кількість двохзначних чисел, які не змінюють свою суму цифр при множенні числа на однозначне ціле число N ($N=0..9$).

Вхідні дані

Ціле число N . $0 \leq N \leq 9$.

Вихідні дані

Відповідь до задачі.

Вхідні дані	Вихідні дані
2	10

Пояснення: При множенні двоцифрових чисел на 2 не змінюють суму цифр такі числа: 18 27 36 45 54 63 72 81 90 99.

Розв'язання

Потрібно перебрати всі двозначні числа та знайти добутки цих чисел на задане число. Знайти суму цифр (a) двозначного числа та суму цифр (b) отриманого числа (цикл while). Якщо суми однакові, то збільшуємо лічильник.

```
var j, i, a, b, k, n: integer;
begin
  readln (n); {вводимо число на яке будуть множиться всі двозначні числа}
  k:=0;
  for i:=10 to 99 do {цикл от 10 до 99 (усі двозначні числа)}
  begin
    j:=i*n; {для зручності задаємо змінну j – добуток двохзначного
    числа i заданого нами однозначного}
    a:= i div 10 + i mod 10; {сума цифр двозначного числа}
    while j<>0 do {сума цифр числа j (у якому кількість цифр може бути
    більше за два)}
    begin
      b:=b+(j mod 10);
      j:=j div 10;
    end;
    if a=b then k:=k+1; {якщо суми цифр двох чисел однакові то лічильник
    збільшуємо на 1}
  end;
  writeln(k); {виводимо кількість чисел}
end.
```

ПЕРЕТВОРЕННЯ

Візьмемо деяке натуральне число N . Будемо змінювати його наступним чином: якщо число парне, то розділимо його на 2, якщо непарне, додамо 1. Після декількох таких змін ми завжди отримаємо число 1. Наприклад, із числа

11 отримується число **12**, потім **6, 3, 4, 2** і, нарешті, **1**. Таким чином, для отримання **1** з **11** потрібно виконати **6** перетворень.

Напишіть програму, яка зчитує натуральне число і виводить кількість перетворень даного числа до отримання **1**.

Вхідні дані

Число **N** ($1 \leq N \leq 10^9$).

Вихідні дані

Кількість перетворень.

Вхідні дані	Вихідні дані
11	6

Розв'язання

n - число яке будемо змінювати, **k**- кількість перетворень. Цикл (**while**) працює до того моменту поки **n**≠1. Для визначення парності скористаємось функцією **mod** (остача від ділення). Якщо число парне, то ділимо його на 2, у протилежному випадку додаємо 1, при кожному виконанні перетворень лічильник збільшуємо на 1.

```
var k,n:longint;  
begin  
  read(n);  
  k:=0;  
  while n<>1 do  
  begin  
    if (n mod 2=0) then n:=n div 2  
    else n:=n+1;  
    k:=k+1;  
  end;  
  writeln(k);  
end.
```

ГУРТОК ХОРОВОГО СПВУ

У деякому начальному закладі функціонує гурток хорового співу. Початок гуртка завжди відбувається однаково: за сигналом керівника гуртка всі **N** учасників стають у коло і кожен **M**-й для розспіву співає гаму.

Керівник гуртка помітив, що розім'яти горло не завжди вдається всім учасникам гуртка. За заданими **N** і **M** допоможіть йому визначити, чи в черговий раз у розминці будуть брати участь всі учасники хору.

Вхідні дані

Вхідні дані складаються з декількох тестових випадків. Кожен тестовий випадок розміщено в окремому рядку і містить два цілих числа **N** і **M**. ($1 \leq N, M \leq 10^3$).

Вихідні дані

Для кожного тестового випадку в окремому рядку виведіть "YES", якщо в розминці приймуть участь усі учасники хору, у протилежному випадку виведіть "NO".

Вхідні дані	Вихідні дані
4 1	YES
6 3	NO

Розв'язання

Рішення задачі зводиться до знаходження найбільшого спільного дільника (НСД). Для цього можна скористатися алгоритмом Евкліда і якщо НСД дорівнює одиниці, то в розминці візьмуть участь усі учасники хору. У програмі використаний цикл **while not (eof) do**, який зчитує дані поки не кінець файлу.

```
var n,m,nod:int64;
begin
while not(eof) do
begin
readln(n,m);
while n * m <> 0 do {алгоритм Евкліда для знаходження НСД}
if n > m then n:= n mod m
else m:= m mod n;
nod:= n + m;
if nod=1 then writeln('YES')
else writeln('NO');
end;
end.
```

ЦІКАВИЙ ДОБУТОК

Визначити всі можливі значення добутку $i*j$, якщо цілочислові значення змінних i та j змінюються відповідно i від a до b та j від c до d ($1 \leq a, b, c, d \leq 10$).

Вхідні дані

У єдиному рядку 4 числа через пропуск: a , b , c та d .

Вихідні дані

Єдине число - кількість можливих варіантів добутку.

Вхідні дані	Вихідні дані
1 10 1 10	42

Розв'язання

Для знаходження всіх можливих значень добутку без повторень $i*j$ будемо перевіряти кожне число p (яке знаходиться в діапазоні від добутку мінімальних значень i та j до добутку їх максимальних значень) на те, чи можливо його отримати за допомогою даного добутку. Якщо отримати число хоча б один раз, перемноживши i та j , реально, то значенню **flag** присвоюємо 1. Якщо **flag** дорівнює 1, то збільшуємо лічильник варіантів добутку (**count**).

```
var a,b,c,d,p,ppoch,pkin,i,j,ipoch,ikin,jpoch,jkin,flag,inti,intj,count:integer;
begin
  readln(a,b,c,d);
  if (a<b)then begin ipoch:=a; ikin:=b; end {знаходимо
  початкове(мінімальне)і кінцеве(максимальне) число проміжку a;b}
  elsebegin ipoch:=b; ikin:=a; end;
  if (c<d) then begin jpoch:=c; jkin:=d; end {знаходимо
  початкове(мінімальне)і кінцеве(максимальне) число проміку c;d}
  else begin jpoch:=d; jkin:=c; end;
  ppoch:=ipoch*jpoch; {мінімальне значення добутку}
  pkin:=ikin*jkin; {максимальне значення добутку}
  count:=0;
  for p:=ppoch to pkin do {перебір усіх значень добутку}
  begin
    flag:=0;
    for inti:=ipoch to ikin do {перебір усіх можливих варіантів добутку}
    for intj:=jpoch to jkin do
      if p=(inti*intj) thenflag:=1; {перевірка чи можливо отримати число з
      проміжку(від мінімального значення до максимального) перемноживши
      inti*intj}
      if flag=1 then count:=count+1; {якщо знайшлось хоча б одне число, то
      збільшуємо лічильник}
    end;
  writeln(count)
end.
```

Задачі з використанням підпрограм та масивів

Ідеї розв'язань задач

Ліпіна В.П., учителя математики та інформатики Прилуцької гімназії № 1 ім. Георгія Вороного, учителя-методиста

ЦУКЕРКИ

Маленький хлопчик потрапив до казкової країни і побачив там дорогу, вздовж якої розкладено мішки з цукерками. На кожному мішку написана кількість цукерок. Хлопчик може взяти в кожному місці два мішки, що лежать поруч. Яку найбільшу кількість цукерок він може взяти?

Вхідні дані

У єдиному рядку задано спочатку кількість мішків N ($4 \leq N \leq 10000$), а потім N чисел через пропуск – кількість цукерок у кожному мішку (усі числа невід'ємні і не перевищують **1000000**).

Вихідні дані

Вивести єдине шукане число – відповідь до задачі.

Ліміт часу **0.5** секунд

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
8 3 8 5 2 1 7 8 5	28

Розв'язання

Знаходимо суми цукерок у мішках поряд, а потім шукаємо їх максимальну суму.

```
var a:array[1..10000] of longint;  
    n,i,j,max:longint;  
begin  
  read(n);  
  for i:=1 to n do  
    read(a[i]);  
  
  for i:=1 to n do  
    a[i]:=a[i]+a[i+1];  
  dec(n);  
  
  max:=0;  
  for i:=1 to n-2 do  
    begin  
      for j:=i+2 to n do
```

```

begin
  if a[i]+a[j] > max then max:=a[i]+a[j];
end;
end;
writeln(max);
end.

```

КЛАВІАТУРА

Усім відомо, що з часом клавіатура зношується, і клавіші на ній починають прилипати. Звичайно, деякий час таку клавіатуру ще можна використовувати, але для натискання клавіш доводиться застосовувати більшу силу.

При виготовленні клавіатури відразу для кожної клавіші задається кількість натискань, які вона повинна витримати. Якщо знати ці величини для клавіатури, що використовується, то для певної послідовності натискання клавіш можна визначити, які клавіші в процесі їх використання зламаються, а які – ні.

Потрібно написати програму, яка визначає, які клавіші зламаються в процесі заданого варіанту експлуатації клавіатури.

Вхідні дані

Перший рядок містить кількість клавіш n ($1 \leq n \leq 100$) на клавіатурі. Другий рядок містить n цілих чисел – c_1, c_2, \dots, c_n , де c_i ($1 \leq c_i \leq 100000$) – кількість натиснень, які витримує i -та клавіша. Третій рядок містить ціле число k ($1 \leq k \leq 100000$) – загальна кількість натиснень клавіш, і останній рядок містить k цілих чисел p_j ($1 \leq p_j \leq n$) – послідовність натискання клавіш.

Вихідні дані

Вивести n рядків, які містять інформацію про справність клавіш. Якщо i -а клавіша зламалась, то i -ий рядок повинен містити слово "yes" (без лапок), якщо ж клавіша працездатна – слово "no".

Ліміт часу **2** секунди

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
5	yes
1 50 3 4 3	no
16	no
1 2 3 4 5 1 3 3 4 5 5 5 5 5 4 5	no
	yes

Розв'язання

Зчитуємо кількість натиснень, які витримує кожна клавіша та заносимо в масив A . Потім читаємо, яка клавіша натискається, та зменшуємо кількість

натиснень цієї клавіші на 1.

Далі перевіряємо, якщо кількість натиснень, яку витримує клавіша менше нуля, то друкуємо 'yes' інакше 'no'.

```
var a,b:array[1..100000] of longint;  
    n,k,i,j:longint;  
begin  
  read(n);  
  for i:=1 to n do  
    read(a[i]);  
  
  read(k);  
  for i:=1 to k do  
    begin  
      read(b[i]);  
      dec(a[b[i]]);  
    end;  
  
  for i:=1 to n do  
    begin  
      if a[i]<0 then writeln('yes')  
        else writeln('no');  
    end;  
end.
```

ХТО ОТРИМАЄ «АВТОМАТ»?

У технічному університеті студенти отримують оцінки за стобальною шкалою, тобто від нуля до ста. Викладач Павло Іванович пообіцяв студентам, що ті, у кого семестровий рейтинг буде вище середнього у групі, отримають "автомат". А староста групи Петька Петров ще й пообіцяв кожному "автоматнику" по шоколадці (у Петьки тато займається постачанням шоколаду у магазини, і Петька з ним домовився про спонсорську допомогу).

І тепер у Петі проблема – йому потрібно в'яснити, скільки шоколадок він повинен принести в університет. Допоможіть йому в добрій справі. Уявляєте, як добре буде, якщо кожен підприємець буде допомагати успішним студентам!

Напишіть програму, яка буде обчислювати кількість "автоматів" у групі студентів. Дуже може бути, що Ваша програма знадобиться не лише для Петі і його батька, а й для підприємців усієї країни, точніше, для всіх розумних людей, які підтримують успішних студентів!

Вхідні дані

Перший рядок містить кількість студентів n ($n \leq 10000$) у групі. Будемо вважати, що почин батька Петра буде взято на озброєння всім прогресивним людством, підприємці будуть боротись за право дати нагороду студентам не лише однієї групи, але й усього університету. Тому кількість студентів велика.

Наступні рядки містять n цілих чисел - рейтинги студентів. Числа відокремлені пропусками і, можливо, символами переведення рядка.

Вихідні дані

Введіть кількість шоколадок, яку Петро повинен принести в університет.

Ліміт часу **1** секунда

Ліміт використання пам'яті **256** MiB

Вхідні дані	Вихідні дані
Sample 1	Sample 1
4	2
10 20 30 40	
Sample 2	Sample 2
10	4
40 50 55 60 55 54 70 88 49 77	

Розв'язання

Зчитуємо кількість студентів та заносимо їх рейтинги в масив A , одночасно шукаючи суму рейтингів. Визначаємо середній бал та рахуємо кількість студентів, у яких рейтинг вище середнього.

```
var a:array[1..10000] of longint;  
    n,s,k,i:longint;  
begin  
  read(n);  
  for i:=1 to n do  
    begin  
      read(a[i]);  
      s:=s+a[i];  
    end;  
  
  s:= s div n;  
  k:=0;  
  for i:=1 to n do  
    begin  
      if a[i]>s then inc(k);  
    end;  
  writeln(k);  
end.
```

ЗМІЙКА

Напишіть програму, яка виводить елемент із рядка X та стовпця Y матриці розміром $N \times M$, яку заповнено наступним чином:

0	1	2	3
7	6	5	4
8	9	10	11

Вхідні дані

Задано натуральні числа N, M, X, Y ($1 \leq X \leq N \leq 50; 1 \leq Y \leq M \leq 50$). N – кількість рядків матриці, M - кількість стовбців матриці, X та Y – номери рядка та стовпця шуканого елемента.

Вихідні дані

Потрібно вивести елемент із рядка X та стовпця Y .

Ліміт часу 1 секунда

Ліміт використання пам'яті 64 МіВ

Вхідні дані	Вихідні дані
5 2 3 1	4

Розв'язання

Основна задача правильно заповнити таблицю. Заповнюємо таблицю змійкою та виводимо значення заданої клітинки.

```
var a:array[1..50,1..50] of longint;  
    n,m,x,y,i,j,k:longint;  
begin  
  read(n,m,x,y);  
  
  k:=0;  
  i:=1;  
  while i<=n do  
  begin  
    for j:=1 to m do  
    begin  
      a[i,j]:=k;  
      inc(k);  
    end;  
    inc(i);  
  for j:=m downto 1 do  
  begin  
    a[i,j]:=k;  
    inc(k);
```

```

end;
inc(i);
end;
writeln(a[x,y]);
end.

```

КООРДИНАТИ СУСІДІВ

Для клітинки з координатами (x, y) у таблиці розміром $M \times N$ виведіть координати її сусідів. Сусідніми називаються клітинки, які мають спільну сторону.

Вхідні дані

Задано натуральні числа M, N, x, y ($1 \leq x \leq M \leq 10^9, 1 \leq y \leq N \leq 10^9$).

Вихідні дані

У вихідний файл виведіть пари координат сусідів цієї клітинки у довільному порядку.

Ліміт часу 1 секунда

Ліміт використання пам'яті 64 МіВ

Вхідні дані	Вихідні дані
3 3	1 2
2 2	2 1
	2 3
	3 2

Розв'язання

Перевіряємо, щоб координати заданої клітинки не були на межі, бо тоді просто буде менше сусідніх клітинок, та виводимо координати сусідніх клітинок.

```

var m,n,x,y:longint;
begin
read(m,n,x,y);

if x<>1 then writeln(x-1,' ',y);
if y<>1 then writeln(x,' ',y-1);
if x<>m then writeln(x+1,' ',y);
if y<>n then writeln(x,' ',y+1);

end.

```


МОЛОКО ТА ПИРІЖОК

Учням першого класу призначають додаткову склянку молока та пиріжок, якщо першокласник важить менше **30** кг. У перших класах школи навчається n учнів. Склянка молока має об'єм **200** мл, а замовлені упаковки молока – **0,9** л. Визначити кількість додаткових пакетів молока та пиріжків, необхідних щодня.

Вхідні дані

У першому рядку задано ціле число n ($0 < n \leq 100$). У наступному рядку знаходяться n додатних чисел – маси кожного першокласника.

Вихідні дані

В одному рядку вивести два цілих числа – кількість додаткових пакетів молока та пиріжків, необхідних щодня.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
20 30 37 31 25 32 29 35 40 28 25 30 34 26 23 20 22 21 30 38 33	2 9

Розв'язання

Зчитуємо вагу учня та підраховуємо скільки учнів важать менше 30 кг. Потім вираховуємо кількість пакетів молока, яка потрібно, урахувавши, що для 1, 2, 3, 4 стаканів молока потрібен цілий пакет. Виводимо кількість пакетів молока та пиріжків.

```
var a:array[1..100000] of real;
    n,k,i,p:longint;
begin
read(n);
k:=0;
for i:=1 to n do
begin
read(a[i]);
if a[i]<30 then inc(k);
end;
p:=(k*2) div 9;
if (k*2) mod 9 >0 then inc(p);
writeln(p, ' ',k);
end.
```

Ідеї розв'язань задач

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста

МІНЕР

Задано список мін. Потрібно скласти поле для гри в мінера.

Вхідні дані

Задано числа **N** і **M** (цілі, додатні, не перевищують **32**) – кількість рядків та стовбців у поля відповідно, далі число **W** (ціле, невід'ємне, не більше **100**) – кількість мін на полі, далі йде **W** пар чисел, координати міни на полі (перше число – рядок, друге число – стовпчик).

Вихідні дані

Потрібно вивести на екран поле. Формат виведення вказано в прикладі.

Ліміт часу **1** секунда

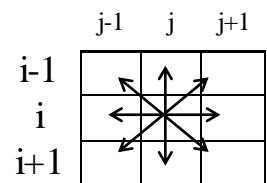
Ліміт використання пам'яті **64** MiB

Вхідні дані	Вихідні дані
Sample 1	Sample 1
3 2	* 2
2	2 *
1 1	1 1
2 2	
Sample 2	Sample 2
2 2	0 0
0	0 0

Розв'язання

У подібних задачах зручно використовувати масиви, що мають додаткові рядки і стовпчики на початку та в кінці. Тому створюємо квадратну таблицю розміром не 32×32 а 34×34 .

Кожна клітинка, яка не містить міни, повинна містити число, що відповідає кількості мін, розміщених у сусідніх клітинках. Для довільної клітинки $a[i,j]$ сусідніми будуть 8 клітинок, у яких і потрібно перевірити наявність міни (обчислити сумарну кількість мін у сусідніх клітинках).



Вказану суму потрібно знайти для кожної клітинки таблиці, для чого можна скористатися подвійним циклом.

```
{Pascal}
var i,j,n,m,w,x,y,k:integer;a:array[0..33,0..33]of integer;
begin
```

Для зручності краще

```
for i:=0 to 33 do
```

```

for j:=0 to 33 do
  a[i,j]:=0;
read(n,m,w);
for i:=1 to w do
  begin
    read(x,y);
    a[x,y]:=-1;
  end;
for i:=1 to n do
  for j:=1 to m do
    if a[i,j]<>-1 then
      begin
        k:=0;
        if a[i-1,j-1]=-1 then k:=k+1;
        if a[i-1,j]=-1 then k:=k+1;
        if a[i-1,j+1]=-1 then k:=k+1;
        if a[i,j-1]=-1 then k:=k+1;
        if a[i,j+1]=-1 then k:=k+1;
        if a[i+1,j-1]=-1 then k:=k+1;
        if a[i+1,j]=-1 then k:=k+1;
        if a[i+1,j+1]=-1 then k:=k+1;
        a[i,j]:=k;
      end;
  end;
for i:=1 to n do
  begin
    for j:=1 to m do
      if a[i,j]=-1 then write('* ') else write(a[i,j], ' ');
    writeln;
  end;
end.

```

СЕРЕДНІЙ БАЛ ДЛЯ ФІГУРИСТІВ

Спортсменам-фігуристам n суддів виставляють оцінки. Технічний працівник змагань вилучає максимальну та мінімальну оцінки, а для залишених оцінок обчислює середнє арифметичне значення. Цей результат вважається балом, що отримав спортсмен. Знайти такий бал для кожного спортсмена.

Вхідні дані

У першому рядку міститься два цілих числа: кількість суддів n та кількість спортсменів m . У наступних m рядках міститься n цілих чисел – оцінки усіх суддів ($0 < n \leq 10$, $0 < m \leq 100$) для кожного з фігуристів.

Вихідні дані

В єдиному рядку вивести m чисел із точністю до двох десяткових знаків –

бал кожного спортсмена.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** МіВ

Вхідні дані	Вихідні дані
5 4	8.33 5.33 9.00 8.50
7 8 9 8 10	
6 5 5 4 7	
9 9 10 7 7	
7 7 10 9 8	

Розв'язання

За умовою не потрібно враховувати мінімальну та максимальну оцінки. Але таких оцінок може бути декілька! Тому маємо або впорядкувати таблицю, або просто першим циклом знайти мінімальну та максимальну оцінку, а другим шукати середній бал.

```
{Pascal}
var i,j,n,m,min,max,k,s:integer;a:array[1..100]of integer;
begin
  read(n,m);
  for i:=1 to m do
    begin
      for j:=1 to n do read(a[j]);
      min:=32767;max:=0;
      for j:=1 to n do
        begin
          if a[j]>max then max:=a[j];
          if a[j]<min then min:=a[j];
        end;
      k:=0;s:=0;
      for j:=1 to n do
        if (a[j]<>min)and(a[j]<>max) then begin s:=s+a[j];k:=k+1;end;
      write(s/k:0:2,' ');
    end;
end.
```

Ідея розв'язання задачі

Хрол Н. П., учителя інформатики загальноосвітньої спеціалізованої школи І-ІІІ ст.. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

МАКСИМУМ

Ваша задача дуже проста і навіть без великої історії: просто потрібно знайти максимум на відрізку.

Вхідні дані

Спочатку подається кількість чисел n ($1 \leq n \leq 10^5$) у масиві. У наступному рядку задано n чисел – вихідний масив a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$). Наступний рядок містить кількість запитів q ($1 \leq q \leq 5 \cdot 10^5$). Кожний із наступних q рядків містить по два натуральні числа l та r ($1 \leq l, r \leq n$) – відрізок, на якому слід знайти максимум.

Вихідні дані

Для кожного запиту виведіть максимум на заданому відрізку.

Ліміт часу **1.5** секунда

Ліміт використання пам'яті **128** МiВ

Вхідні дані	Вихідні дані
10	10
5 1 2 8 7 6 10 7 5 6	10
8	8
1 10	8
5 10	2
1 5	10
2 6	10
2 3	10
7 7	
7 8	
5 9	

Розв'язання

Якщо використати базовий алгоритм пошуку максимального елемента в масиві, то тестуючи, система зараховує тільки 58% тестів. 42% тестів перевищують ліміт часу, відведений на виконання програми.

Збережемо початковий масив A в масив C та індекси елементів масиву A в масив B .

$B[i]$	1	2	3	4	5	6	7	8	9	10
$A[i]$	5	1	2	8	7	6	10	7	5	6

Впорядкуємо масив А (наприклад, за зростанням) таким чином, щоб при перестановці елементів масиву А переставлялися відповідно елементи масиву В.

V[i]	2	3	1	9	6	10	5	8	4	7
A[i]	1	2	5	5	6	6	7	7	8	10

Після зчитування чергової пари лівої і правої межі відрізка (змінні l та r) будемо розглядати один із двох випадків:

1. Якщо відстань між лівою і правою межею не більше 1, то достатньо порівняти C[l] і C[r] та вивести більший із них;
2. Інакше переглянути масив В: якщо $l \leq B[j] \leq r$, то вивести A[j].

Лістинг програми:

```

var A,B,C:array[1..100000] of int64;
    n,l,r,q,k:int64;
    i,j:longint;
procedure qsort(l,r:longint);
    var e,f1,f2:int64;
    i,j:longint;
    begin
        i:=l;
        j:=r;
        e:=A[(i+j) div 2];
        while i<=j do
            begin
                while A[i]<e do inc(i);
                while A[j]>e do dec(j);
                if i<=j then begin
                    f1:=A[i];
                    A[i]:=A[j];
                    A[j]:=f1;
                    f2:=B[i];
                    B[i]:=B[j];
                    B[j]:=f2;
                    inc(i);
                    dec(j);
                end;
            end;
        if l<j then qsort(l,j);
        if i<r then qsort(i,r);
    end;

```

```

begin
assign(input,'input.txt');
assign(output,'output.txt');
reset(input);
rewrite(output);
readln(n);
for i:=1 to n do
  begin
  read(A[i]);
  B[i]:=i;
  end;
readln(q);
c:=a;
qsort(1,n);
for i:=1 to q do
  begin
  readln(l,r);
  if abs(l-r)<=1 then begin
    if c[l]>c[r] then writeln(c[l]) else writeln(c[r]);

    end else begin
  if l>r then begin
    k:=l;
    l:=r;
    r:=k;
    end;
  for j:=n downto 1 do
    if (B[j]>=l) and (B[j]<=r) then
      begin
        writeln(A[j]);
        break;
      end;end;

  end;
close(input);
close(output);
end.

```

Список використаних джерел

1. Білоусова Л.І. Краса простих задач або до питання про використання мов програмування у навчанні школярів інформатики // Комп'ютер у школі та сім'ї. – 2014. - №1. – С.18-22

2. Бондаренко С.М. Збірник задач та розв'язків II етапу Всеукраїнської учнівської олімпіади з інформатики 2014-2015 навчального року / С.М. Бондаренко, В.В. Зуб, Ю.М. Літош. – Чернігів: ЧОППЮ імені К.Д. Ушинського, 2015. – 35 с.

3. Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України [Електронний ресурс]. – Режим доступу: <http://e-olymp.com/> - Назва з екрану.

4. Чернігівська обласна Інтернет-школа «Юний програміст» [Електронний ресурс]. – Режим доступу: <http://choippo.cn.sch.in.ua/> - Назва з екрану.