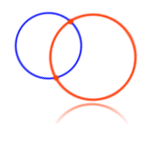
<https://www.e-olymp.com/uk/problems/4>

Два кола

Визначити в скількох точках перетинаються два кола.



**Вхідні дані**

6 чисел x1, y1, r1, x2, y2, r2, де x1, y1, x2, y2, - координати центрів кіл, r1, r2 – їх радіуси. Всі числа - дійсні, не перевищують 1000000000 за модулем, та задані не більш ніж з 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 #1**

0 0 5 5 0 1

**Вихідні дані #1**

2

Пояснення – Все робить теорема Піфагора ( радіус – гіпотенуза, різниця координат центрів – катети)

1. Накладання –(-1)

(x1==x2 && y1==y2 && r1==r2)

1. Дотикання – 1

((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)==(r1+r2)\*(r1+r2)

1. Дотикання всередині -1

((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)==(r1-r2)\*(r1-r2))

1. Немає спільних точок

((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)>(r1+r2)\*(r1+r2))

((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)<(r1-r2)\*(r1-r2)

1. Дві точки

((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)<(r1+r2)\*(r1+r2) &&

(x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)>(r1-r2)\*(r1-r2))

#include <iostream>

using namespace std;

int main()

{

double x1, y1, r1, x2, y2, r2;

cin>>x1>>y1>>r1>>x2>>y2>>r2;

if (x1==x2 && y1==y2 && r1==r2) cout<<-1; else

if ((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)==(r1+r2)\*(r1+r2)) cout<<1; else

if ((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)==(r1-r2)\*(r1-r2)) cout<<1; else

if ((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)>(r1+r2)\*(r1+r2)) cout<<0; else

if ((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)<(r1-r2)\*(r1-r2)) cout<<0; else

//cout<<2;

if ((x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)<(r1+r2)\*(r1+r2) &&

(x1-x2)\*(x1-x2)+ (y1-y2)\*(y1-y2)>(r1-r2)\*(r1-r2)) cout<<2;

cout << endl;

return 0;

}

<https://www.e-olymp.com/uk/problems/8287>

Петро підприємець

Петро приватний підприємець і він продає різні цукерки. Петро помітив, що деякі цукерки шалено популярні, а інші взагалі не користуються попитом.

В голові приватного підприємця виникла ідея зробити асорті (змішати два види цукерок - популярні і не популярні). Взявши різну масу кожного виду цукерок Петро отримав асорті вартість **1** кг якого **А** грн.

Знаючи, що популярні цукерки коштують **P** грн/кг а не популярні **N** грн/кг, а також значення **А**, знайдіть скільки грам популярних цукерок в асорті.

**Вхідні дані**

Три дійсних числа **P**, **N**, **А** ціна **1** кг різних видів цукерок, що входять до складу асорті,та ціна асорті.

**Вихідні дані**

Одне дійсне число округлене до десятих - кількість грамів популярних цукерок в асорті, або **-1** якщо визначити не можливо.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

100 50 75

**Вихідні дані #1**

500.0

<https://www.e-olymp.com/uk/problems/3217>

Новий Рік

Новий Рік в Кріляндії приходить несподівано в квітні, а також у червні, вересні і грудні. На черговий Новий Рік крілики Брайан і Стью отримали багато цукерків. Вони зібрали всі свої цукерки у дві коробки і вирішили дізнатися, скільки різних видів цукерок є одночасно в коробках Стью і Брайана.

**Вхідні дані**

Перший рядок містить два цілих числа, відокремлених пропуском: **N**, **M** (**1** ≤ **N**, **M**≤ **1000**). Другий рядок містить **N** назв цукерок з коробки Стью, відокремлених пропусками, а третій рядок – аналогічно **M** назв цукерок з коробки Брайана. Кожна назва цукерок складається не більше ніж з двадцяти літер (малі латинські від **a** до **z**) і не менше ніж з однієї.

**Вихідні дані**

Вивести одне число – кількість різних видів цукерок, що є і в коробці Стью, і в коробці Брайана.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані**

5 5

snikers murs skitless murs murs

baunti snikers baunti murs murs

**Вихідні дані**

2

<https://www.e-olymp.com/uk/problems/8293>

Трикутна спіраль (trispiral)

На площині з точки **(0;0)** провели відрізок прямої до точки **(1;2)**. З точки **(1;2)** провели відрізок прямої до точки **(3;-2)**. З точки **(3;-2)** провели відрізок прямої до точки **(-2;-2)**. З точки **(-2;-2)** провели відрізок прямої до точки **(1;4)**. Далі за такими правилами будується трикутна спіраль. За значеннями координат довільної точки **x, y** вивести **-1**, якщо вона не є однією з кутових точок (вершин) спіралі, а якщо є – визначити номер **n** цієї вершини. Відрахунок номерів починається від точки **(0;0)**, номер якої нуль, і ведеться у напрямку руху стрілки годинника по спіралі.

**Вхідні дані**

Ввести два цілих числа: **x, y**. (**-108 ≤ x, y ≤ 108**).

**Вихідні дані**

Вивести номер вершини на спіралі **n** або **-1**.

<https://www.e-olymp.com/uk/problems/8288>

Олімпіада з програмування

**Олімпіада з програмування**

На АСМ-олімпіаду прибуло **N** учасників. В результаті анкетування члени журі встановили, що **A** учасників кодять на Ci, **B** на Python, **C** на Pascal, **X** одночасно знають Ci та Python, **Y** - Python та Pascal, **Z** - Ci та Pascal. Маючи значення **N, A, B, C, X, Y, Z** встановіть кількість учасників, що програмують на трьох мовах програмування.

**Вхідні дані.**

В одному рядку через пробіл сім дійсних чисел значення **N, A, B, C, X, Y, Z** значення яких не перевищують **100**.

**Вихідні дані**

Одне єдине значення - кількість учасників, що програмують на трьох мовах програмування.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

100 40 50 60 15 20 25

**Вихідні дані #1**

10

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

1 4

**Вихідні дані #1**

4

**Вхідні дані #3**

200000 20000

**Вихідні дані #3**

-1

<https://www.e-olymp.com/uk/problems/8289>

Прості паліндроми.

На проміжку від **a** до b підрахувати кількість простих паліндромів. Паліндромом називається число яке читається рак само в зворотному порядку, наприклад **7, 22, 121**. Простим називається число, яке має лише **2** дільники.

**Вхідні дані.**

В одному рядку через пропуск записано два числа **a** та **b**. **(0 ≤ a,b ≤ 107)**

**Вихідні дані**

Одне єдине значення - кількість простих паліндромів.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

5 12

**Вихідні дані #1**

3

<https://www.e-olymp.com/uk/problems/8263>

Різні цифри

Сєні подобаються числа, в яких немає двох однакових цифр підряд. Задано ціле число **n**. Допоможіть Сєні знайти мінімальне ціле число, строго більше **n**, яке йому подобається.

**Вхідні дані**

Вхідні дані містять ціле число **n (1 ≤ n ≤ 1018)**.

**Вихідні дані**

Виведіть мінімальне ціле число більше **n**, в якого немає двох однакових цифр підряд.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

98

**Вихідні дані #1**

101

<https://www.e-olymp.com/uk/problems/8262>

# Підрахунок в строю

Весь рік Вася не ходив до університету, тому не здав екзамени, і його відрахували. Так він потрапив до армії. А одне із самих популярних занять в армії — стояти в строю.

У Васиному взводі **n** солдат, рахуючи його. Солдати стоять в одну шеренгу, кожен із них дивиться або вліво, або вправо, а також має свій порядковий номер від **1** до **n**, що дорівнює його місцю в шерензі. Ріст **i**-го солдата дорівнює **hi**. Вася вважає, що солдат з номером **i** бачить солдата з номером **j**, якщо виконується наступні умови:

* солдат **i** дивиться в сторону солдата **j**;
* всі солдати, що стоять між ними, не вище солдата **j**.

Так, наприклад, якщо в шерензі стоять **4** солдата ростом **h1 = 178**, **h2 = 180**, **h3 = 170**, **h4 = 190**, а також всі солдати дивляться вліво, то **2**-й солдат буде бачити тільки **1**-го, **3**-й — тільки **2**-го (так як між ним і першим є більш високий другий солдат), **4**-й буде бачити **2**-го і **3**-го солдат.

Так як зайнятися в строю все одно нічим, Вася хоче порахувати, скільки людей бачить кожен із солдат.

#### Вхідні дані

Перший рядок вхідних даних містить число **n** — кількість солдат в шерензі **(1 ≤ n ≤ 105)**.

Другий рядок містить **n** чисел **h1; h2; ... ; hn** — ріст солдат в шерензі **(1 ≤ hi ≤ 109)**.

Третій рядок містить **n** символів, що описує напрямок, в який дивляться солдати:

**i**-й символ дорівнює **«L»**, якщо **i**-й солдат дивиться вліво, тобто може побачити тільки солдат з номерами **1; 2; ... ; i - 1**, або **«R»**, якщо **i**-й солдат дивиться вправо може побачити тільки солдат з номерами **i + 1; i + 2; ... ; n**.

#### Вихідні дані

Виведіть **n** цілих чисел, **i**-те з виведених чисел повинне дорівнювати кількості солдат, яких бачить **i**-й солдат в строю.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

4

178 180 170 190

LLLL

**Вихідні дані #1**

0 1 1 2

**Вхідні дані #2**

5

178 180 175 170 190

LLRLL

**Вихідні дані #2**

0 1 2 2 3

**Вхідні дані #3**

5

178 180 170 170 160

LLRLL

**Вихідні дані #3**

0 1 1 2 3

|  |
| --- |
| #include "iostream"  using namespace std;  int main()  {long int N, A[100000], n,smax;  char B[100000];  cin>>N;  for(long i=0;i<N;i++){cin>>A[i];}  cin>>B;  for(int i=0;i<N;i++){n=0;  if(B[i]=='L' && i>0)  {n=1;  smax=A[i-1];  for(long j=(i-2); j>=0; j--)  {if(A[j]>=smax){n++;smax=A[j];}  //if(n>0 && A[j]<A[j+1] && A[j+1] !=A[i])n--;  }  }  else  if(B[i]=='R' && i<N-1)  {  n=1;  smax=A[i+1];  for(long j=(i+2); j<N;j++){  if(A[j]>=smax){n++;smax=A[j];}  //if(n>0 && A[j]<A[j-1] && A[j-1] != A[i])n--;  }  }  cout<<n<<" ";  }  return 0;  } |

<https://www.e-olymp.com/uk/problems/7433>

# Система числення

Вам задане число А та основа системи числення P – обидва в десятковому представленні. Знайдіть запис числа А у P-ковій системі числення.

**Вхідні дані**

Перший рядок вхідних даних містить десятковий запис числа А довжиною не більше 1000 знаків. Другий рядок містить десятковий запис основи системи числення P, також довжиною не більше 1000 знаків.

**Вихідні дані**

Виведіть у вихідний рядок запис числа А у P-ковій системі числення, причому, якщо цифра у P-ковій системі числення знаходиться у межах від 0 до 9, виводьте саму цифру, якщо ж цифра є більшою за 9, її десятковий запис необхідно взяти в квадратні дужки (дивись приклад).

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

123

21

**Вихідні дані #1**

5[18]

<https://www.e-olymp.com/uk/problems/2385>

# Кількість перестановок

За заданим натуральним числом **n** знайти кількість різних перестановок чисел від **1** до **n**.

**Вхідні дані**

Одне число **n** (**1** ≤ **n** ≤ **12**).

**Вихідні дані**

Вивести кількість різних перестановок чисел від **1** до **n**.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані**

3

**Вихідні дані**

6

<https://www.e-olymp.com/uk/problems/390>

# Анаграми

Анаграмою слова називається довільна перестановка всіх літер слова. Наприклад, зі слова **SOLO** можна отримати **12** анаграм: **SOLO**,**LOSO**, **OSLO**, **OLSO**, **OSOL**, **OLOS**, **SLOO**, **LSOO**, **OOLS**, **OOSL**, **LOOS**, **SOOL**.

Напишіть програму, яка виводить кількість різних анаграм, які можна отримати з цього слова.

#### Вхідні дані

Слово, кількість літер в якому не перевищує **14**.

#### Вихідні дані

Кількість різних анаграм.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

SOLO

**Вихідні дані #1**

12

<https://www.e-olymp.com/uk/problems/60>

# Площа многокутника

Задано координати **n** послідовних вершин многокутника. Знайти його площу.

#### Вхідні дані

Перший рядок містить кількість вершин многокутника **n** (**3** ≤ **n** ≤ **50000**). У наступних **n** рядках задано цілочисельні координати його послідовних вершин xi, yi (**-1000** ≤ xi, yi ≤ **1000**).

#### Вихідні дані

Вивести площу многокутника з трьома десятковими знаками.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

3

0 0

0 2

2 0

**Вихідні дані #1**

2.000

<https://www.e-olymp.com/uk/problems/2148>

# Опуклий многокутник

Визначити, чи є заданий многокутник опуклим.

**Вхідні дані**

У першому рядку одне число - **N** (**3** ≤ **N** ≤ **100000**). Далі у **N** рядках по парі чисел - координати чергової вершини простого многокутника у порядку обходу за або проти годинникової стрілки. Усі координати цілі числа, які по модулю не перевищують **10000**.

**Вихідні дані**

Один рядок "**YES**", якщо наведений многокутник є опуклим, і "**NO**" у протилежному випадку.

<https://www.e-olymp.com/uk/problems/678>

Працівники

На заводі кожна з **N** деталей може бути обробленою на одному з двох верстатів: **A** або **B**. Кожна деталь має порядковий номер від **1** до **N**. До обробки деталі поступають послідовно, у відповідності зі своїми номерами. Кількість деталей завжди парна.

Існують правила, за якими визначається чи можна обробляти деталь на певному верстаті.

1. Якщо на поточний момент на верстаті **B** була оброблена така ж кількість деталей, як і на верстаті **A**, то наступна деталь повинна бути оброблена на верстаті **A**.
2. У підсумку на кожному з верстатів повинно бути оброблено однакову кількість деталей.

Скільки існує людей, стільки і думок. Кожен із працівників цього заводу запропонував свою послідовність обробки деталей, причому всі пропозиції виявилися різними, але такими, що задовольняють правилам **1** і **2**.

Напишіть програму, що за інформацією про кількість деталей N визначає максимальну можливу кількість працівників заводу.

**Вхідні дані**

Одне парне число **N** (**2** ≤ **N** ≤ **28**) - кількість деталей, яку необхідно обробити.

**Вихідні дані**

Вивести одне ціле число - максимальну можливу кількість працівників заводу.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані**

4

**Вихідні дані**

2

<https://www.e-olymp.com/uk/problems/2479>

Баланс дужок

Є рядок, що містить дужки **( )** та **[ ]**. Дужковий вираз вважається правильним, якщо:

* він є порожнім
* якщо **A** та **B** правильні, то **AB** правильно
* якщо **A** правильно, то (**A**) та [**A**] правильні

Напишіть програму, яка за вхідним рядком, що містить дужковий вираз, визначить його правильність. Довжина рядка не більша за **128**символів.

**Вхідні дані**

Перший рядок містить кількість тестів **n**. Кожний з наступних **n** рядків містить вираз, який складається із дужок **( )** та **[ ]**.

**Вихідні дані**

Для кожного тесту вивести в окремому рядку "**Yes**", якщо вираз є правильним і "**No**" інакше.

Ліміт часу **1** секунда

Ліміт використання пам'яті **122.49** MiB

**Вхідні дані #1**

3

([])

(([()])))

([()[]()])()

**Вихідні дані #1**

Yes

No

Yes

# <https://www.e-olymp.com/uk/problems/4680>

# Делители

За заданим натуральним числом **n** обчислити кількість натуральних чисел, які є дільниками **n!** (факторіала числа **n**).

Наприклад, при **n** = **4**, **n!** = **4** \* **3** \* **2** \* **1** = **24**. Це число має наступні дільники: **1**, **2**, **3**, **4**, **6**, **8**, **12**, **24**. Таким чином, шукана кількість становить **8**.

#### Вхідні дані

Одне ціле число **n** (**1** ≤ **n** ≤ **45**).

#### Вихідні дані

Вивести одне ціле число – кількість дільників числа **n!**

Ліміт часу 1 секунда

Ліміт використання пам'яті 122.17 MiB

**Вхідні дані**

4

**Вихідні дані**

8

# <https://www.e-olymp.com/uk/problems/223>

# Еволюція

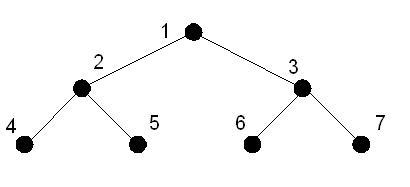
Під час досліджень, присвячених появі життя на планеті Олімпія, вченими було зроблено декілька сенсаційних відкриттів:

1. Усі живі організми планети походять від однієї бактерії Bitozoria Programulis.

2.Еволюція проходила крок за кроком (за припущенням вчених – під час змін клімату на планеті).

3.На кожному кроці еволюції з кожного виду утворювалися рівно два підвиди, а попередній вид зникав.

4.Якщо вважати появу бактерії Bitozoria Programulis першим кроком еволюції, то нині існуючі живі організми знаходяться на **n**-му кроці.



Щоб не вигадувати назви під час досліджень, вчені пронумерували всі види організмів, що будь-коли існували на планеті. Для цього вони намалювали дерево еволюції із коренем Bitozoria Programulis, яка отримала номер **1**. Далі нумерували види кожного кроку еволюції зліва направо. Таким чином безпосередні підвиди Bitozoria Programulis отримали номери **2** та **3**. Наступними були занумеровані види третього кроку еволюції – підвиди виду **2** отримали номери **4** та **5**, а виду **3 -**номери **6**та **7**, і т.д.

Напишіть програму, яка за номерами двох видів обчислить номер виду їх найближчого спільного предка у дереві еволюції.

**Вхідні дані**

Перший рядок містить кількість етапів еволюції **n**(**1**≤**n**≤**100**), що відбулися на планеті Олімпія до теперішнього часу. Другий та третій рядки містять по одному натуральному числу, що представляють номери видів, для яких потрібно знайти номер їх найближчого спільного предка.

**Вихідні дані**

Вивести одне натуральне число - номер найближчого предка для двох видів.

Ліміт часу 1 секунда

Ліміт використання пам'яті 122.17 MiB

**Вхідні дані #1**

4

15

12

**Вихідні дані #1**

3

**Вхідні дані #2**

18

233016

233008

**Вихідні дані #2**

14563

# <https://www.e-olymp.com/uk/problems/987>

# Цвяхи

На прямій дощечці вбито цвяхи. Довільні два цвяхи можна з'єднати ниточкою. Потрібно з'єднати деякі пари цвяшків ниточками так, щоб до кожного цвяха була прив'язана хоча б одна ниточка, а сумарна довжина усіх ниточок була мінімальною.

#### Вхідні дані

У першому рядку кількість цвяхів **n** (**1** ≤ **n** ≤ **100**). У наступному рядку записано **n** чисел - координати усіх цвяхів (невід'ємні цілі числа, які не перевищують **10000**).

#### Вихідні дані

Вивести мінімальну сумарну довжину всіх ниточок.

Ліміт часу 1 секунда

Ліміт використання пам'яті 122.17 MiB

**Вхідні дані #1**

5

4 10 0 12 2

**Вихідні дані #1**

6

Сначала отсортируем гвоздики по возрастанию координат. Решим следующую подзадачу: найдем минимальную длину веревочек, необходимую для того, чтобы связать первые k гвоздиков согласно условию (обозначим требующуюся для этого длину веревочек ak).

Можно считать, что любая ниточка связывает два соседних гвоздика (иначе ее можно разрезать на несколько частей, связывающих все гвоздики между теми, которые связывала наша ниточка изначально).

Научимся вычислять ak. Заметим, что в оптимальной конфигурации (для первых k гвоздиков) между последним (k-м) и предпоследним ((k-1)-м) гвоздиками ниточка есть всегда, а вот между предпоследним ((k-1)-м) и предпредпоследним ((k-2)-м) она может либо быть, либо не быть. В первом случае первые k-1 гвоздиков удовлетворяют условию задачи, во втором - первые k-2. Значит ak=min(ak-1,ak-2)+lk-1,k, где lk-1,k - расстояние между k-м и k-1-м гвоздиками (в отсортированном массиве). Для удобства вычислений удобно ввести фиктивные первый и нулевой элементы равные 0 и бесконечности соответственно (в реальной программе в роли бесконечности обычно выступает какое-нибудь достаточно большое число, например для данной задачи вполне подойдет 30000). Теперь последовательно заполняя массив a с помощью данной формулы, мы получим верный ответ на поставленную задачу в элементе aN.

Число действий, выполненных данным алгоритмом, пропорционально N.

#include "iostream"

#include "algorithm"

using namespace std;

long int n,a[1000],b[1000];

int main()

{

cin>>n;

for(int i=1;i<=n;i++) cin>>a[i];

/\*for (int i=0;i<n-1;i++)

for (int j=0;j<n-1;j++)

if (a[j]>a[j+1])

swap(a[j],a[j+1]);

\*/

sort(a+1,a+n+1);

//for(int i=1;i<=n;i++) cout<<a[i]<<" ";cout<<endl;

b[0]=0;

b[1]=3000000;

for (int i=2; i<=n;i++)

b[i]=min(b[i-1],b[i-2])+a[i]-a[i-1];

//for(int i=0;i<=n;i++) cout<<b[i]<<" ";cout<<endl;

cout<<b[n]<<endl;

return 0;

}