**1. Нулі на кінці n!**

Час на тест: 1 секунда.

Оцінка: 100 балів.

За даним числу n визначте, якою кількістю нулів закінчується десяткова запис числа n! .

Формат вхідних даних

Програма отримує на вхід єдине натуральне число n, 1≤ n <= 10^9 і повинна вивести єдине натуральне число.

приклад

Вхідні дані:

      100

Вихідні дані:

      24

*#include <fstream>*

*using namespace std;*

*ifstream cin("input9.txt");*

*ofstream cout("output9.txt");*

*int main() {*

*long long int n;*

*cin>>n;*

*long long int res = 0;*

*while (n) {*

*res += n / 5;*

*n /= 5;*

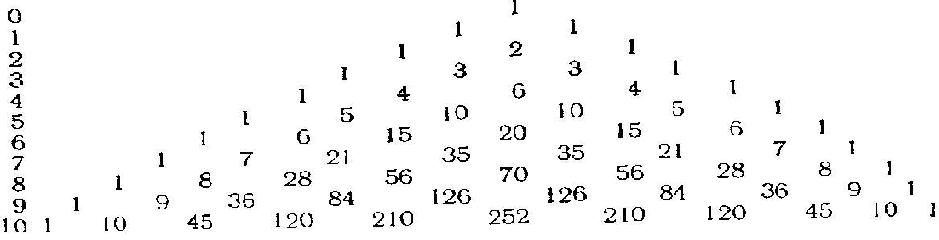
*}*

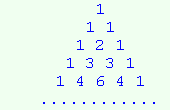
*cout<<res<<endl;*

*return 0;*

*}*

**Задача 2**





Обчислити суму N рядків трикутника Паскаля (1<=N<=100).

1,2,4,8,16

2^3-1=7

2^4-1=1+2+4+8=15

http://subject.com.ua/master/Algebra/lesson19-image001.gif

#include <cstdlib>

#include "iostream"

#include "fstream"

using namespace std;

int main ()

{

int n,i,j;

int a[1000],c[1000];

int os;

ifstream cin ("text.in");

ofstream cout("text.out");

cin>>n;

cin.close();

a[0]=1;

a[1]=1;

for (j=1 ;j<=n;j++)

{

os=0;c[0]=a[0];

for(i=1;i<=c[0];i++)

{

c[i]=(a[i]\*2+os)%10;

os=(a[i]\*2+os)/10;

}

while (os>0){c[0]=c[0]+1;c[c[0]]=os;os=os/10;}

for (i=0;i<=c[0];i++)a[i]=c[i];

}

a[1]=a[1]-1;

for (i=a[0];i>=1;i--)

cout<<a[i];

cout<<"\n";

cout.close();

}

**Задача 3**

Припустимо, щороку Ви кладете s в банк під v % річних. Скільки грошей Ви матимете на рахунку через t років?

Формула ---------------------- 2,000 · 1.056 = 2680.19

**Задача 4**

Час на тест - 30 секунд.

У деякій державі в обігу знаходяться банкноти певних номіналів. Національний банк хоче, щоб банкомат видавав будь-яку запитану суму за допомогою мінімального числа банкнот, вважаючи, що запас банкнот кожного номіналу необмежений. Допоможіть Національному банку вирішити цю задачу.

Формат вхідних даних

Перший рядок вхідних даних містить натуральне число n, 0 <n <= 100. Другий рядок вхідних даних містить n різних натуральних чисел x1, x2, ..., xn, не переважаючих 1.000.000. Третій рядок містить натуральне число S, не перевершує 5.000.000.

Формат вихідних даних

Програма повинна знайти представлення числа S у вигляді суми доданків з множини {xi}, що містить мінімальне число доданків і вивести це подання на екран (у вигляді послідовності чисел, розділених пробілами). Якщо таких уявлень існує декілька, то програма повинна вивести будь-яке (одне) з них. Якщо таке подання не існує, то програма повинна вивести слово impossible.

Приклад

Вхідні дані

     5

     1 3 7 12 32

     40

Вихідні дані

     1 7 32

Вхідні дані

     5

     10 50 100 500 1000

     99

Вихідні дані

     Impossible

#include <cstdlib>

#include "iostream"

#include "fstream"

using namespace std;

int main ()

{ifstream cin ("input0.txt");

ofstream cout("output0.txt");

int n,k,a[100],b[100];

int m, i;

cin>>k;

for (i=0;i<k;i++)cin>>a[i];

cin>>n;

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

const int INF=1000000000;// Значення константи "нескінченність"

int F[1000];

F[0]=0;

for(m=1;m<=n;++m) // заповнюємо масив A

{ // m - сума, котору видати

F[m]=INF; // відмічаємо, що суму i видати не можна

for(i=0;i<k;++i) // перебираємо всі номінали банкнот

{

if(m>=a[i] && F[m-a[i]]+1<F[m])

F[m] = F[m-a[i]]+1;// змінюємо значення F[m], якщо знайшли

} // кращий спосіб видати суму m

}

int j=0;

if (F[n]==INF)

cout<<"Impossible"<<endl;

else

{

while(n>0)

{

for(i=0;i<k;++i)

{

if(F[n-a[i]]==F[n]-1)

{

j++;

b[j]=a[i];

n=n-a[i];

break;

} } } }

for (i=1;i<j;i++)cout<<b[i]<<" "; cout<<b[j]<<endl; }

**Задача 5.**

**(московські олімпіади 2013,** [**http://olympiads.ru/moscow/2012-13/index.shtml**](http://olympiads.ru/moscow/2012-13/index.shtml)**)**

Колекціонер дуже любить ювелірні вироби. Його колекція нараховує багато алмазів, перлин. В тому числі і безцінний рубін з діамантом Кліопатри, добутий два тижні назад. А от сьогодні у нього появився шанс купити легендарні коштовності.

На цей раз на чорний ринок попали K золотих монет з відбитком Гая Юлія Цезаря. Колекціонер не поскупився і викупив зразу N монет (N ≥ K), лишні з яких є звичайно фальшивими.

В колекціонера є ваги, котрі показують сумарну вагу покладених на них предметів. Знаючи, що справжня монета, повинна важити а грам, а фальшива, зроблена невмілими спекулянтами, b грам. Поможіть колекціонеру визначити мінімальне число зважувань, необхідне для знаходження справжніх монет.

Перший рядок вхідного файлу містить числа T - кількість тестів в файлі.

Наступні T рядів містить по 4 цілих числа N, K,a,b (1≤N≤100,0≤K≤N, 1≤a,b≤1000).

Для кожного тесту знайти мінімальне число зважувань, необхідне для виявлення всіх дорогоцінних монет. Знайдені числа розмістіть в вихідному файлі по одному в рядку.

**Додаткові задачі**

**Задача 6. Календар**

Час на тест: 1 секунда.

Програма отримує на вхід два рядки, що містять дати, записані у форматі dd.mm.yyyy, де dd - день місяця від 01 до 31, mm - номер місяця від 01 до 12, yyyy - номер року від 0001 до 2999. Програма повинна обчислити різницю між цими датами (кількість днів між ними) і вивести відповідь у вигляді натурального числа. Обидві дати коректні, друга пізніше першої.

Врахуйте, що роки, чиї номери поділяються на 4 і не діляться на 100, а також всі роки, чиї номери поділяються на 400 є високосними.

**Приклад**

**Вхід Вихід**

01.01.0001 1

02.01.0001

**Вхід Вихід**

29.02.2004 366

01.03.2005

**Задача 7. Морський бій**

На прямокутному полі для гри в морський бій розміром M × N розташовано декілька прямокутних кораблів. Кораблі не стикаються один з одним. Ваше завдання - визначити всілякі типи кораблів на поле і число кораблів кожного типу. Два корабля відносяться до одного типу, якщо їх розміри співпадають (кораблі, які можуть бути отримані один з одного поворотом, також відносяться до одного типу).

**Вхідні дані**

Перший рядок вхідних даних містить два позитивні числа M і N, не перевершують 1000, що задають розміри поля. Далі йде M рядків, кожен з яких складається з N символів. Символ `1' означає, що відповідна клітка поля зайнята кораблем, символ ` 0' - що вільна. Пропусків рядку немає.

**Вихідні дані**

Програма повинна для кожного виявленого типу корабля вивести один рядок, що містить три числа. Перші два числа задають розміри корабля (перше число повинно бути не менше другого), третє число задає кількість кораблів даного типу на полі. Рядки у виведенні повинні бути відсортовані по першому числу, потім по другому числу.

|  |  |
| --- | --- |
| **Приклад вхідних даних**  6 10  0111000011  0000011011  0100011000  0101011011  0100000000  0001111011 | **Приклад вихідних даних**  1 1 1  2 1 2  2 2 2  3 1 2  3 2 1  4 1 1 |

**Задача 8. Муха**

Час на тест: 1 секунда.

На поверхні прямокутного паралелепіпеда сидить муха і нанесена крапля варення. Визначте найменшу відстань, яке повинна проповзти муха по поверхні паралелепіпеда, щоб доповзти до краплі.

**Формат вхідних даних**

Перші три рядки вхідних даних містять позитивні числа K, L, M, є розмірами паралелепіпеда уздовж осей OX, OY, OZ. Один кут паралелепіпеда знаходиться на початку координат, протилежний - в точці (K, L; M), ребра паралелепіпеда паралельні осям координат.

Наступні три рядки вхідних даних містять координати мухи X1, Y1, Z1, потім йдуть три рядки з координатами варення X2, Y2, Z2. Задаються цими координатами точки знаходяться на поверхні паралелепіпеда. Всі числа у вхідних даних цілі, не перевершують 1000.

**Формат вихідних даних**

Програма повинна вивести єдине число - найкоротша відстань, яке повинна проповзти муха по поверхні паралелепіпеда, щоб дістатися до варення. Відповідь необхідно вивести у вигляді дійсного числа, перевірка здійснюватиметься з точністю 10-3.

**Приклад**

|  |  |
| --- | --- |
| **Вхід Вихід**  1 2.236067977  1  1  0  0  0  1  1  1 | **Вхід Вихід**  12 13  5  2  1  0  1  12  4  1 |

**Задача 9. Завод**

На підприємстві планується виконати 8 робіт: v1, v2, ... v8. Для виконання цих робіт необхідні механізми а1, а2, ..., а6. Використання механізмів для проведення кожної з робіт визначається наступною таблицею Т:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Робота  механ. | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 |
| a1 | + |  | + |  |  |  | + | + |
| a2 |  | + |  | + |  |  |  |  |
| a3 |  |  | + |  |  | + | + |  |
| a4 | + | + |  | + | + |  |  |  |
| a5 |  |  | + |  | + |  |  | + |
| a6 |  |  |  |  | + | + |  | + |

Ніякої з механізмів не може бути зайнятий одночасно на двох роботах. Всі роботи виконуються за одне і теж час t. Як розподілити механізми, щоб сумарний час виконання всіх робіт було найменшим?

**Задача 10. Проводи**

Є N точок і M проводів. Дротом можна з'єднати деяку пару різних точок, причому пара може бути з'єднана не більше ніж одним дротом. Усі проводки повинні бути використані.

Нехай Di - кількість дротів, які будуть з'єднані з точкою з номером i, i = 1, ..., N.

Необхідно поєднати N точок з допомогою M проводков таким чином, щоб сума S = D1 \* D1 + D2 \* D2 + ... + Dn \* Dn була максимальною.

Вивести величини Di в неспадному порядку і. на вимогу

(priznak = 1), список з'єднань.

ENTER:

<Введіть N:> N (N <= 100)

<Введіть M:> M (M <= 1000)

<PRIZNAK=> PRIZNAK

ВИСНОВОК:

<Результуюча конфігурація:> Di в неспадному порядку.

<Сума S> S

<Список з'єднань>

<Крапку 1 з'єднати з> список точок

.....

<Крапку N з'єднати з> список точок