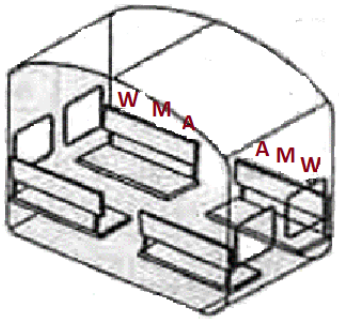
Шацков Иван Александрович

Балл: 99

Задача №1

Условие

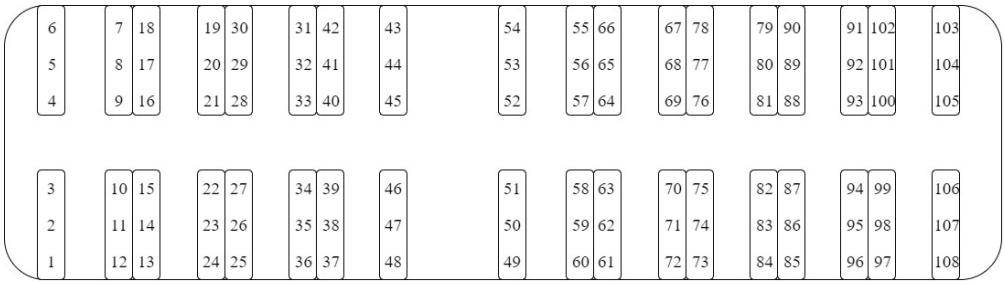
Юля и **Петя** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



Тип места обозначается следующим образом:

- Сиденье у окна: **W**
- Среднее место: **M**
- Место у прохода: **A**

Нумерация мест в вагоне приведена на следующем рисунке



Ребятам стало интересно узнать номер и тип сиденья, расположенного рядом с ними. В случае, когда сиденье расположено в середине, необходимо указать оба соседних номера.

Напишите программу, которая поможет ребятам решить эту задачу.

Входные данные

На вход подается одно целое число **N (**1 <= **N** <= 108**),** обозначающее номер места.

Выходные данные

Целое число, и большая латинская буква **W, M** или **A**, разделенные одним пробелом, которые будут соответствовать номеру и типу соседнего сиденья. Если таких сидений два, то данные по второму сидению выводится в той же строке через пробел.

Примеры

Входные данные	Выходные данные
19	17 M
29	31 W 33 A

Исходный код

```
#include<iostream>
#include<conio.h>
#include<string>
#include<math.h>
using namespace std;
int main (){
    string p[109];
    int i,n,m,k=1;
    for (i=1;i<=108;i++){
    if (k==1||k==6)
    p[i]="W";
    if (k==2||k==5)
    p[i]="M";
    if (k==3||k==4)
    p[i]="A";
    if (k<=5) k++;
    else k=1;
    }
    cin>>n;


    if (p[n]=="M")
    cout<<n-1<<" "<<p[n-1]<<" "<<n+1<<" "<<p[n+1];
    else
    if (p[n]=="A"&&p[n+1]=="M")
    cout<<n+1<<" "<<p[n+1];
    else
    if (p[n]=="A"&&p[n-1]=="M")
    cout<<n-1<<" "<<p[n-1];
    else
    if (p[n]=="W"&&p[n-1]=="M")
    cout<<n-1<<" "<<p[n-1];
    else
    if (p[n]=="W"&&p[n+1]=="M")
    cout<<n+1<<" "<<p[n+1];

    getch();

}
```

Задача №2

Условие

Вася любит придумывать миры для настольных игр. При рисовании политической карты очередного мира он задается вопросом: можно ли раскрасить эту карту с помощью заранее известного количества цветов так, чтобы никакие два соседних государства не оказались окрашены одним цветом?

Напишите программу, которая решит эту задачу за Васю.

Сначала на вход программы подаются два натуральных числа: количество цветов n, в которые можно красить карту, $1 < n \le 5$, и количество стран на карте k, $1 < k \le 8$. Далее на вход программы подаются k строк по k чисел – матрица смежности стран. j-ое число i-ой равно единице, если между i-ой и j-ой странами есть общая граница и ноль, если общей границы у них нет. Считается, что сама с собой страна не граничит.

Выведите одно целое число – сколькими способами можно раскрасить карту.

Пример

Входные данные	Выходные данные

2 3 0 1 0 1 0 1 0 1 0	2

Комментарий к примеру.

В распоряжении Васи два цвета, на его карте три страны. Страны №1 и №2 граничат только со страной №2. Соответственно, можно покрасить страны №1 и №3 в цвет №1, а страну №2 в цвет №2, можно наоборот, покрасить страны №1 и №3 в цвет №2, а страну №2 – в цвет №1.

Исходный код

```
var
n,n1,k,s,c0,c1,i,j,n2:integer;
a: array [1..8,1..8] of integer;
begin
readln(n,k);
c0:=0;
c1:=0;
for i:=1 to k do
for j:=1 to k do
begin
read(a[i,j]);
if a[i,j]=0 then c0:=c0+1;
if a[i,j]=1 then c1:=c1+1;
end;
if c0=k*k then begin
n1:=1;
while k<>0 do
begin
n1:=n1*n;
k:=k-1;
end;
end
else
if c1=k*k then
if n=k then n1:=n
else if n<k then n1:=0
else
begin
n1:=n-1;
while n1>1 do
begin
n:=n*n1;
n1:=n1-1;
end;
n2:=n-k-1;
while n2>1 do
begin
n:=n*n2;
n2:=n2-1;
end;
n1:=n1 div n2;
end
else n1:=n;
writeln(n1);
end.
```

Задача №3

Условие

Исполнитель Септенариум получает на вход натуральное число X (не превышающее 10^6). По этому числу, точнее по его представлению в семеричной системе счисления строится новое число Y по следующим правилам.

В семеричном представлении числа X зачеркиваются последние две цифры (гарантируется, что в семеричном представлении числа X больше 2-х цифр).

1. Если получившееся число нечетное, то в конец полученной цепочки цифр дописывается две цифры –

сумма оставшихся цифр (тоже в семеричном представлении, причем если сумма состоит из одной цифры, то перед ней надо вписать 0).

Например, $2305_{10} = 6502_7$. После зачеркивания остается 65_7 – нечетное число. Сумма оставшихся цифр $11_{10}=14_7$. Новое число $Y = 6514_7 = 2314_{10}$.

Другой пример, $841_{10} = 2311_7$, после зачеркивания остается нечетное число 23_7. Сумма оставшихся цифр $5_{10} = 5_7$. Новое число $Y= 23\mathbf{0}5_7 = 838_{10}$.

2. Если получившееся после зачеркивания последних двух цифр число четное, то в конец полученной цепочки цифр дописывается одна цифра – количество оставшихся цифр. Например, $2302_{10} = 6466_7$. После зачеркивания остается 64_7 – четное число. Количество оставшихся цифр $2_{10}=2_7$. Новое число $Y = 642_7 = 324_{10}$.

Напишите программу, которая будет искать наибольшее число, которое получается на заданном интервале [A,B], и наименьшее число, из которого оно было получено.

На вход программы подаётся два целых числа A и B ($50 \le A \le B \le 1\ 000\ 000$), записанных через пробел.

Программа должна вывести два числа наибольшее полученное число и через пробел число, из которого оно было получено исполнителем.

Входные данные	Вывод	Примечание
2302 2305	2314 2303	2302 -> 324 **2303 -> 2314** 2304 -> 2314 2305 -> 2314 Наибольшее значение, получаемое на заданном интервале 2314. Оно получается из трех чисел 2303, 2304 и 2305. Наименьшее из них 2303.

Исходный код

```
#include<iostream>
#include<conio.h>
#include<math.h>
using namespace std;
int cc(int n){
    int k=1,x=0;
    while (n!=0)
    {
        x+=(n%7)*k;
        n/=7;
        k*=10;

    }return(x);
}
int main(){
int a,b,n,i,m,k,p,s=0,x,l=1,x1=0;
bool f=true;
cin>>a>>b;
m=cc(b)/100;
k=cc(b)/100;

p=b;
for (i=b-1;i>=a;i-- &&f ){f=false;
    if ((cc(i)/100)==m){

    f=true;
    p=i;}}


while (m>0)
{
    s+=m%10;
```

```
    m/=10;
}
x=k*100+cc(s);
while (x>0){
    x1+=(x%10)*l;
    l*=7;
    x/=10;
}
cout<<x1<<" "<<p;


getch();
}
```

Задача №4

Условие

Передатчик один раз в секунду отправляет сигнал, состоящий из одного слова. Слово представляет собой последовательность латинских строчных букв. При нормальной работе передатчика длины соседних слов отличаются. Время от времени с передатчиком происходит сбой. Когда с передатчиком происходит сбой, он начинает передавать случайные слова такой же длины, как последнее слово, отправленное нормально. После исправления сбоя передатчик отправляет слово, длина которого отличается от длины слова, после которого начался сбой.

Напишите программу, которая найдет самый долгий сбой.

Формат ввода

В первой строке вводится сначала целое число ***n*** – количество слов (***n ≤ 1000***), затем в ***n*** следующих строках записано по слову. Слова состоят только из строчных латинских букв.

Формат вывода

Вывести одно целое число – сколько секунд длился самый долгий сбой.

Примеры

Входные данные	Выходные данные
3 abc kda msbv	1
4 bca bdc abc abcfde	2

Исходный код

```
var
n,i,l,k,m:integer;
s:string;
begin
readln(n);
l:=0;
m:=0;
for i:=1 to n do
begin
readln(s);
if length(s)=l then
```

```
k:=k+1
else begin
if k>=m then m:=k;
k:=0;
l:=length(s);
end;
end;
if k>=m then m:=k;
write(m);
end.
```

Задача №5

Условие

Юстас передает Алексу шифровку, состоящую из четырехзначных чисел. В каждом числе последняя цифра подобрана так, чтобы число делилось на 3. Центр принял шифровку Юстаса вперемешку с сообщением от Вихря, состоящим из натуральных чисел. Известно, что в сообщении Вихря ни одно число не бывает одновременно четырехзначным и делящимся на три. Напишите программу, которая определит, сколько чисел переданы Юстасом.

Формат ввода

В строке вводится сначала целое число ***n*** – количество чисел в шифровке (***n ≤ 1000***), затем ***n*** натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – количество переданных Юстасом чисел.

Пример

Входные данные	Выходные данные
4 1002 2000 333 123	1

Исходный код

```
#include<iostream>
#include<conio.h>
#include<math.h>
using namespace std;
int main(){
    int n,k=0,a,i;
    cin>>n;
    for (i=1;i<=n;i++){
        cin>>a;
        if (a>=1000 &&a<=9999&&a%3==0)
            k++;}
        cout<<k;
        getch();
}
```

Задача №6

Условие

Для обогрева пассажирского отсека вездехода «Химера» необходимо подключить нагревательный элемент. Элемент питается от сети постоянного тока напряжением U и состоит из двух последовательно соединенных резисторов, сопротивление каждого из которых не должно быть меньше порогового значения R. Напишите программу, которая найдет в списке резисторов пару, которая обеспечит наибольшую мощность элемента. Гарантируется, что хотя бы два подходящий нагревателя в списке есть.

На вход программе в первой строке подаются натуральное число N – количество резисторов (N≤20) и вещественное положительное число R – ограничение по мощности.

Далее в N строках подается на вход по целому положительному числу – r_i: сопротивление i-го резистора.

Вывести одно целое число – сопротивление нагревательного элемента.

Мощность рассчитывается по формуле

$$P = \frac{U^2}{R}$$

где U – напряжение, R – сопротивление. При последовательном соединении сопротивления складываются.

Пример

Ввод	Вывод
3 5 5 6 7	11

Исходный код

```
#include<iostream>
#include<conio.h>
#include<math.h>
using namespace std;
int main(){
    int i,j,n,r[20],m1,m2,m;
    float R;
    cin>>n>>R;
    for (i=0;i<n;i++)
        cin>>r[i];
        m=r[0];
    for (i=0;i<n;i++)
        if (r[i]>m)
        m=r[i];
    m1=m;
    m2=m;

    for(i=2;i<n;i++){
        if (r[i]<m1 && r[i]>=R)
        {
            m2=m1;
            m1=r[i];
        }
        else if (r[i]<m2 && r[i]>=R)
            m2=r[i];
    }
    cout<<m1+m2;
    getch();


}
```

Задача №7

Условие

В электрической цепи требуется собрать заданное сопротивление, используя как можно меньше резисторов. В распоряжении сборщика находится несколько типов резисторов, которые подключают последовательно. Для каждого из типов резисторов известно его сопротивление в омах.

Напишите программу, которая по заданным характеристикам резисторов и заданном сопротивлении найдет минимальное количество резисторов, с помощью которых можно собрать цепь.

На вход программе в первой строке подаются два натуральных числа R ($R \leq 1000$) и k ($k \leq 8$) – заданное сопротивление и количество типов резисторов. Далее в k строках идет натуральное число r_i – сопротивление типа резистора под номером i. Количество резисторов каждого типа считается неограниченным.

Вывести одно целое число – минимальное количество резисторов, достаточное для сбора цепочки. Если из резисторов нельзя собрать цепочку, вывести 0.

Пример

Ввод	Вывод
11 3 3 2 1	4

Исходный код

```
var
r,n,k,i,j,x,r1,l:integer;
a,s: array[1..8]of integer;
begin
read(r,n);
r1:=r;
for i:=1 to n do begin
read(a[i]);
s[i]:=0;
end;
for i:=1 to n do
for j:=1 to n-i do
if a[j]<a[j+1] then
begin
x:=a[j];
a[j]:=a[j+1];
a[j+1]:=x;
end;
for i:=1 to n do begin
l:=i;
r1:=r;
while (r1>0) and (l<=n) do begin
while r1 >=0 do begin
r1:=r1-a[l];
s[i]:=s[i]+1;
end;
if r1<0 then begin
r1:=r1+a[l];
s[i]:=s[i]-1;
end;
l:=l+1;
end;
if r1<>0 then s[i]:=0;
end;
x:=2000;
for i:=1 to n do
if (s[i]< x)and (s[i]<>0) then
x:=s[i];

writeln(x);


end.
```