

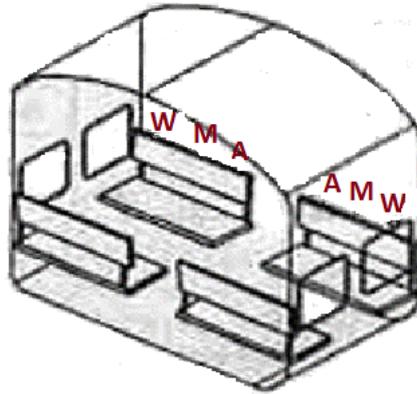
**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по программированию «Профессор Лебедев»
(общеобразовательный предмет информатика), весна 2020 г.**

9 класс

Вариант 1

Задача 1

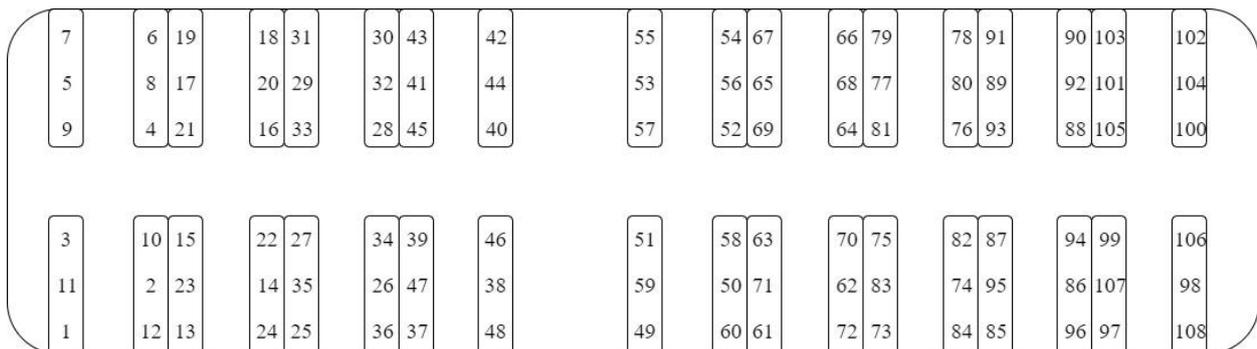
Натasha и Олег очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



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 ($1 \leq N \leq 108$), 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 выводятся в той же строке через пробел (номера сидений в этом случае выводятся в порядке возрастания).

Примеры

Входные данные	Выходные данные
19	18 M
29	32 M

Тесты

Входные данные	Выходные данные
19	18 M
29	32 M
96	85 W
33	28 A
24	13 W
38	47 M
66	67 W
47	38 M
12	1 W
4	9 A

Решение

```
seats = ['W', 'M', 'A']
m = int(input())
temp = m - 1
coupe = temp // 12
temp %= 12
side = temp // 6
temp %= 6
part = (temp // 3) ^ side
temp %= 3
seat_ar = 12 * (coupe + 1) - (m - 12 * coupe) + 1
if side == part:
    print('{} {}'.format(seat_ar, seats[temp]))
else:
    print('{} {}'.format(seat_ar, seats[2 - temp]))
```

Задача 2

В кассе, продающей билеты по цене 500 р., нет возможности расплатиться картой, а в начале работы у кассира для сдачи есть только m купюр по 500 р. Перед открытием у кассы собрались посетители, у каждого человека в очереди есть только одна купюра. У части – 500 р., у части – 1000 р. Сколькими способами можно выстроить посетителей в очередь так, чтобы к моменту обслуживания посетителя с купюрой в 1000 рублей у кассира всегда была сдача?

Напишите программу, которая решит эту задачу.

На вход программы подаются три целых неотрицательных числа:

n - количество человек с купюрой 500 р,

k - количество человек с купюрой 1000 р

m - количество купюр по 500 р у кассира к началу работы кассы.

Каждое число меньше 10.

Выведите одно целое число – сколькими способами можно выстроить очередь.

Пример

Входные данные	Выходные данные
1 2 1	4

Комментарий к примеру. Пусть есть посетители **A**, **B**, и **C**. У **A** и **B** по 1000 рублёвой купюре, а у **C** – купюра в 500 рублей. Тогда их можно выстроить так:

ABC – нельзя, т.к. единственная купюра 500р. уйдет на сдачу посетителю **A**, а для посетителя **B** сдачи не будет.

ACB – можно, т.к. единственная купюра 500р. уйдет на сдачу посетителю **A**, затем посетитель **C** заплатит купюрой 500р., и для покупателя **B** будет сдача.

BAC – нельзя, т.к. единственная купюра 500р уйдет на сдачу посетителю **B**, а для посетителя **A** сдачи не будет.

BCA – можно

CAB – можно

CBA – можно

Есть только 4 возможных варианта.

Тесты

Входные данные	Выходные данные	Баллы
1 2 1	4	0
3 3 0	180	10
6 6 1	222393600	10
5 6 0	0	10
5 6 1	11404800	10
5 6 2	25660800	10
6 3 3	362880	10
6 2 4	40320	10
9 9 9	6402373705728000	10
9 8 0	71137485619200	10
9 7 1	12934088294400	10

Решение

```

n1, n0, m = map(int, input().split())
f = [[0] * (n1 + 2 + m) for i in range(n1 + n0 + 1)]
f[0][m] = 1
for l in range(1, n1 + n0 + 1):
    for b in range(n1 + 1 + m):
        f[l][b] = f[l - 1][b - 1] + f[l - 1][b + 1]
res = f[n1 + n0][m + n1 - n0] if n1 + m >= n0 else 0
for i in range(2, n0 + 1):
    res *= i
for i in range(2, n1 + 1):
    res *= i
print(res)

```

Задача 3

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

В восьмеричном представлении числа X предпоследняя цифра увеличивается на 1 (гарантируется, что в восьмеричном представлении X числа больше 2-х цифр). Например, $695_{10} = 1267_8 \rightarrow 1277_8 = 703_{10}$.

Если предпоследняя цифра 7, тогда предпоследняя цифра становится 0, а последняя изменяется по следующему принципу: четная увеличивается на 1, а нечетная уменьшается на 1. Например, последняя цифра нечетная $697_{10} = 1271_8 \rightarrow 1200_8 = 640_{10}$, последняя цифра четная $698_{10} = 1272_8 \rightarrow 1203_8 = 643_{10}$.

Введем понятие расстояния

$$Oh = | \text{Исходное_число} - \text{Полученное_число} |$$

Напишите программу, которая будет считать наибольшее расстояние Oh для чисел из заданного интервала $[A, B]$ и наибольшее исходное число, для которого оно было вычислено.

На вход программы подаётся два целых числа A и B ($10 \leq A \leq B \leq 1\,000\,000$), записанных через пробел.

Программа должна вывести два числа наибольшее расстояние Oh и через пробел исходное число, для которого оно было посчитано.

Входные данные	Вывод	Примечание
694 698	57 697	$ 694 - 702 = 8$ ($1266_8 - 1276_8$) $ 695 - 703 = 8$ ($1267_8 - 1277_8$) $ 696 - 541 = 55$ ($1270_8 - 1201_8$) $ 697 - 640 = 57$ ($1271_8 - 1200_8$) $ 698 - 643 = 55$ ($1272_8 - 1203_8$)
693 695	8 695	$ 693 - 701 = 8$ ($1265_8 - 1275_8$) $ 694 - 702 = 8$ ($1266_8 - 1276_8$) $ 695 - 703 = 8$ ($1267_8 - 1277_8$)

Тесты

Ввод	Вывод
2356 2360	55 2360
502 506	57 505
4087 4087	8 4087
1914 1914	55 1914
10 1000000	57 999999
999900 1000000	57 999999
1000000 1000000	8 1000000
10 10	8 10
60 60	55 60
9976 9978	57 9977

Решение

```
Program isp3_2;
var x,a,b,k,mx,oh,ox:longint;
const d=8;
function digit(x:longint):longint;
var a,b,c:longint;
begin
  b := x mod d;
  a :=(x mod (d*d))div d;
  c := x div (d*d);
  a := a+1;
  if (a>=d)then
    begin
      a := 0;
      if(b mod 2=0) then b := b+1
      else b := b-1;
    end;
  x:= c*d*d +a*d+ b;
  digit := x
end;
begin
mx:=0;
readln(a,b);
for x:=a to b do begin
  oh:=abs(x-digit(x));
  if oh >= mx then begin
    mx := oh;
    ox := x
  end;
end;
writeln(mx, ' ', ox);
end.
```

Задача 4

Сообщение, которое передают по каналу связи, состоит из чисел, записанных в десятичной системе счисления. Каждое число состоит из шести знаков, а его двоичная запись оканчивается тремя нулями. При передаче сообщение было засорено посторонними шумами: числами, отличающимися от тех, что были в сообщении. Найдите изначальное количество чисел в сообщении.

Формат ввода

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

Формат вывода

Вывести одно целое число – количество достоверных сигналов в сообщении.

Пример

Входные данные	Выходные данные
3 100128 4356 234064	2

Тесты

Входные данные	Выходные данные
3 100128 4356 234064	2
4 234864 123048 230 1000	2
5 123048 547800 234000 123400 345864	5
6 100400 323464 100000 433864 64 32	4
1 100000	1
5 123064 345080 345 32 500800	3
2 40 111400	1
3 40 111400 11400	1
4 40 111400 11400 222800	2
5 40 111400 11400 222800 131072	3

Решение

```
program z83z92v3;
```

```
function good(num:integer):boolean;
```

```
var sum,k:integer;
```

```
begin
```

```
  sum:=0;
```

```
  k:=0;
```

```
  while num > 0 do
```

```
  begin
```

```
    sum:=sum+num mod 10;
```

```
    num:=num div 10;
```

```
    k:=k+1;
```

```
  end;
```

```
  good := (true) and (k=6);
```

```
end;
```

```
var x,m,n,i:integer;
```

```
begin
```

```
  read(n);
```

```
  m:=0;
```

```
  for i:=1 to n do
```

```
  begin
```

```
    read(x);
```

```
    if good(x) and (x mod 8 = 0) then
```

```
      m:=m+1;
```

```
    end;
```

```
  writeln(m);
```

```
end.
```

Задача 5

Несколько агентов пересылают кодовые сообщения в Центр. Сообщение каждого агента представляет собой несколько слов, к каждому из которых приписан его идентификатор, состоящий из одной буквы. Сообщения записаны по очереди. Найдите сообщение, содержащее больше всего слов.

Формат ввода

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

Формат вывода

Вывести одно целое число – длину сообщения (количество слов в сообщении), содержащего больше всего слов.

Примеры

Входные данные	Выходные данные
4 bcd bcfd bcdfe abc	2
3 abd abdc bvd	1

Тесты

Входные данные	Выходные данные
4 bcd bcfd bcdfe abc	2
3 abd abdc bvd	1
5 b bb bbb bbbb bbbbb	5
7 abc abdc f g gh dfh cfh	3
5 a ab c d e	1

8 abv vbav fgh dfge fhn fth aav aab	2
1 asdvbn	1
10 abc abc abc df ad acf abc abc d abc	3
5 a b bb bbb bbbb	4
3 abc abc d	2

Решение

program z85z94v1_1203;

var

n,i,x,len,maxlen:integer;

s,ps:string;

begin

 readln(n);

 maxlen:=0;

 len:=1;

 ps:='';

for i:=1 **to** n **do**

begin

 readln(s);

if copy(ps,length(ps),1) = copy(s,length(s),1) **then**

begin

 len:=len+1;

```

end
else
begin
if len>maxlen then
maxlen:=len;
len:=1;
end;
ps:=s;
end;
if len>maxlen then
maxlen:=len;
writeln(maxlen);
end.

```

Ситуационная задача 1.

Павел ведет дневник, наблюдая за погодой. Каждый день он записывает температуру. Требуется определить, в какой день во время периода наблюдений была наименьшая температура и сколько дней отмечалась температура выше порогового значения

На вход программе в первой строке подаются натуральное число N – количество дней, в течение которых велось наблюдение ($N \leq 20$) и вещественное число T – пороговое значение температуры.

Далее в N строках подается на вход по вещественному числу – t_i : Температура в i -й день.

Вывести два целых числа – в первой строке вывести номер дня, в который отмечалась наименьшая температура, во второй строке вывести, сколько дней отмечалась температура выше порогового значения. Если наименьшая температура наблюдалась в несколько разных дней, вывести наименьший номер дня.

Пример

Ввод	Вывод
3 5	1
5	2
6	
7	

Тесты

Ввод	Вывод
1 1	1
1.1	1
5 0	1
0	0
0	
0	
0.0	
5 0	5
-1	0
-1.1	
-2.1	
-0.3	
-5	

10 2 1.5 1.7 1.9 2.1 2.4 2.3 2.5 4.0 4.5 5.1	1 7
8 -2 -3 -2.4 -4 -4.5 -6 -5.6 -7.2 -1	7 1
3 0 60 55 60	2 3
5 40 39 38 40 41 42	2 2
1 1 1	1 0
3 -1 -1 -1.1 -0.9	2 1
4 -1.5 -1 -1.1 -2 -1.9	3 2

Решение

program z801_1103;

var

n,t,j,k:integer;

t,ti,tmax:real;

begin

 readln(n,t);

 k:=0;

```

tj:=1;
readln(tmax);
if tmax>t then
  k:=k+1;
for j:=2 to n do
  begin
  readln(ti);
  if ti<tmax then
    begin
    tj:=j;
    tmax:=ti;
    end;
  if ti>t then
    k:=k+1;
  end;
  writeln(tj);
  writeln(k);
end.

```

Ситуационная задача 1.

Робот может выполнять команды «Поиск дефектов», «Движение», «Подготовка» и «Ремонт». Из-за конструктивных особенностей на робота наложены некоторые ограничения. Два раза подряд можно выполнить только команду «Движение». Команда «Ремонт» может быть выполнена только на следующем шаге после команды «Подготовка».

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n , если до начала выполнения программы робот выполнил команду «Подготовка».

На вход программе подается натуральное число n ($n \leq 15$) – количество команд.

Вывести целое число – количество выполнимых последовательностей команд длиной n .

Пример

Ввод	Вывод
2	8

Тесты

Ввод	Вывод
1	3
3	22
10	24960
12	186304
15	3799168

Решение

```

program z8102_1103;

```

```

var

```

```

i,n,a,b,c,d,pa,pb,pc,pd:integer;

```

```

begin

```

```

  readln(n);

```

```
a:=0; //Поиск
b:=0; //Движение
c:=1; //Подготовка
d:=0; //Ремонт
for i:=1 to n do
  begin
    pa:=a; pb:=b; pc:=c; pd:=d;
    a:=pb+pa+pd+pc;
    b:=pa+pc+pd;
    c:=pa+pb+pd;
    d:=pc;
  end;
writeln(a+b+c+d);
end.
```

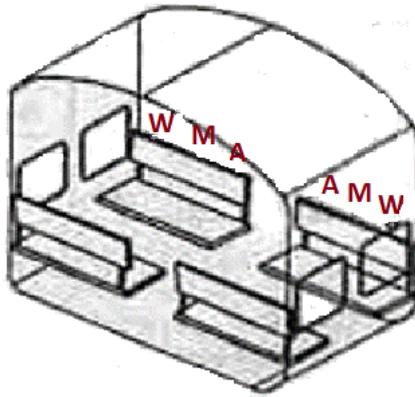
**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по программированию «Профессор Лебедев»
(общеобразовательный предмет информатика), весна 2020 г.**

9 класс

Вариант 2

Задача 1

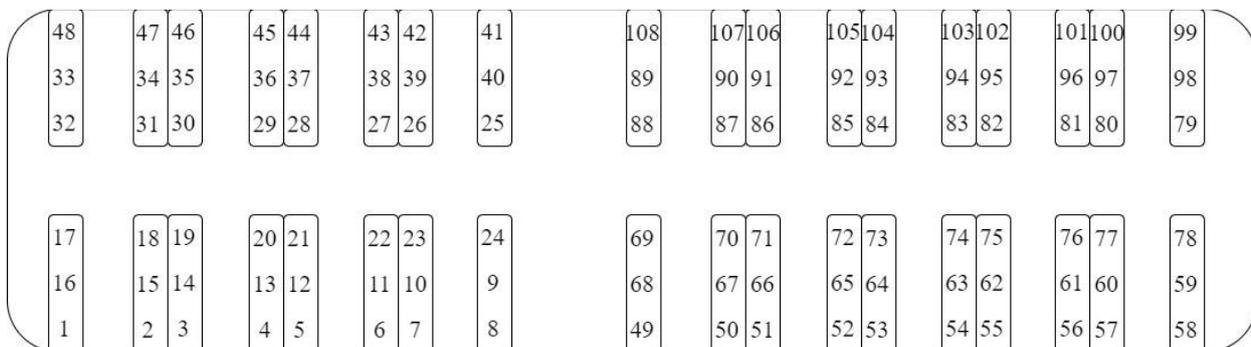
Миша и **Маша** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



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 напротив. Напишите программу, которая поможет ребятам решить эту задачу.

Входные данные

На вход подается одно целое число N ($1 \leq N \leq 108$), обозначающее номер места.

Выходные данные

Целое число, и большая латинская буква **W**, **M** или **A**, разделенные одним пробелом, которые будут соответствовать номеру и типу сиденья, расположенного напротив.

Примеры

Входные данные	Выходные данные
19	20 А
96	95 А

Тесты

Входные данные	Выходные данные
19	20 А
96	95 А
29	30 А
33	34 М
24	23 А
38	37 М
66	65 А
47	48 W
12	11 М
4	3 W

Решение

```
n = int(input())
seat = ""
if n <= 48:
    r = (n - 1) // 8
else:
    r = (n - 1) // 10
if r == 0 or r == 5:
    seat = 'W'
elif r == 1 or r == 4:
    seat = 'M'
else:
    seat = 'A'
print(n + (-1) ** (n % 2 + 1), seat)
```

Задача 2

В кассе, продающей билеты по цене 500 р., нет возможности расплатиться картой, а в начале работы нет даже сдачи. Перед открытием у кассы собрались посетители, у каждого человека в очереди есть только одна купюра. У части – 500 р., у части – 1000 р. Сколькими способами можно выстроить посетителей в очередь так, чтобы к моменту обслуживания посетителя с купюрой в 1000 рублей у кассира всегда была сдача?

Напишите программу, которая решит эту задачу.

На вход программы подаются два целых неотрицательных числа:

n - количество человек с купюрой 500 р,

k - количество человек с купюрой 1000 р

Каждое число меньше 10.

Выведите одно целое число – сколькими способами можно выстроить очередь.

Пример

Входные данные	Выходные данные
2 1	4

Комментарий к примеру. Пусть есть посетители **А**, **В**, и **С**. У **А** и **В** по 500 рублёвой купюры, а у **С** – купюра в 1000 рублей. Тогда их можно выстроить так:

АВС – можно

АСВ – можно

ВАС – можно

ВСА – можно

САВ – нельзя, т.к. в кассе к началу работы нет купюр на сдачу, а у первого посетителя купюра 1000р.

СВА – нельзя, т.к. в кассе к началу работы нет купюр на сдачу, а у первого посетителя купюра 1000р.

Есть только 4 возможных варианта.

Тесты

Входные данные	Выходные данные
2 1	4
3 3	180
8 3	26611200
7 4	19958400
5 6	0
9 1	3265920
9 8	71137485619200
9 9	640237370572800
0 1	0
4 0	24
0 0	1

Решение

```
n1, n0 = map(int, input().split())
f = [[0] * (n1 + 2) for i in range(n1 + n0 + 1)]
f[0][0] = 1
for l in range(1, n1 + n0 + 1):
    for b in range(n1 + 1):
        f[l][b] = f[l - 1][b - 1] + f[l - 1][b + 1]
res = f[n1 + n0][n1 - n0] if n1 >= n0 else 0
for i in range(2, n0 + 1):
    res *= i
for i in range(2, n1 + 1):
    res *= i
print(res)
```

Задача 3

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

В шестеричном представлении числа X предпоследняя цифра увеличивается на 1 (гарантируется, что в шестеричном представлении X числа больше 2-х цифр). Например, $749_{10} = 3245_6 \rightarrow 3255_6 = 755_{10}$.

Если предпоследняя цифра 5, тогда предпоследняя цифра становится 0, а последняя изменяется по следующему принципу: четная увеличивается на 1, а нечетная уменьшается на 1.

Например, последняя цифра нечетная $751_{10} = 3251_6 \rightarrow 3200_6 = 720_{10}$, последняя цифра четная $752_{10} = 3252_6 \rightarrow 3203_6 = 723_{10}$.

Введем понятие расстояния

$Oh = | \text{Исходное_число} - \text{Полученное_число} |$

Напишите программу, которая будет считать наибольшее расстояние Oh для чисел из заданного интервала [A,B] и наибольшее исходное число, для которого оно было вычислено.

На вход программы подаётся два целых числа A и B ($10 \leq A \leq B \leq 1\,000\,000$), записанных через пробел.

Программа должна вывести два числа наибольшее расстояние Oh и через пробел исходное число, для которого оно было посчитано.

Пример

Входные данные	Вывод	Примечание
748 752	31 751	$ 748 - 754 = 6$ ($3244_6 - 3254_6$) $ 749 - 755 = 6$ ($3245_6 - 3255_6$) $ 750 - 721 = 29$ ($3250_6 - 3201_6$) $ 751 - 720 = 31$ ($3251_6 - 3200_6$) $ 752 - 723 = 29$ ($3252_6 - 3203_6$)
747 749	6 749	$ 747 - 753 = 6$ ($3243_6 - 3253_6$) $ 748 - 754 = 6$ ($3244_6 - 3254_6$) $ 749 - 755 = 6$ ($3245_6 - 3255_6$)

Тесты

Ввод	Вывод
1034 1038	29 1038
208 212	31 211
4087 4087	6 4087
1940 1940	29 1940
10 1000000	31 999971
999900 1000000	31 999971
1000000 1000000	6 1000000
10 10	6 10
70 70	29 70
68 70	31 69

Решение

```

Program isp4_2;
var x,a,b,k,mx,oh,ox:longint;
const d=6;
function digit(x:longint):longint;
var a,b,c:longint;
begin
  b := x mod d;
  a :=(x mod (d*d))div d;
  c := x div (d*d);
  a := a+1;
  if (a>=d)then
  begin
    a := 0;
    if(b mod 2=0) then b := b+1
  
```

```

    else b := b-1;
  end;
  x:= c*d*d +a*d+ b;
  digit := x
end;
begin
mx:=0;
readln(a,b);
for x:=a to b do begin
  oh:=abs(x-digit(x));
  if oh >= mx then begin
    mx := oh;
    ox := x
  end;
end;
writeln(mx, ' ', ox);
end.

```

Задача 4

Зонд передает данные с орбиты Юпитера во время сильной магнитной бури. Информация передается по каналу связи в виде пакетов. Каждый пакет представляет собой целое положительное число в двоичной системе счисления. Для обнаружения помех последний разряд в пакете подбирают таким образом, чтобы количество единиц в разрядах пакета было четным. В каждом пакете на практике никогда не искажается больше одного разряда.

Напишите программу, которая по распечатке пакетов, записанных в десятичной системе счисления, найдет самое большое значение, прошедшее без искажений. Известно, что как минимум один пакет прошел без искажений.

Формат ввода

В строке вводится сначала целое число n – количество пакетов ($n \leq 1000$), затем n натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – самое большое значение, прошедшее без искажений.

Пример

Входные данные	Выходные данные
4 1025 496 882 1056	1056

Тесты

Входные данные	Выходные данные
4 1025 496 882 1056	1056
5 1025 4 5 6 7	1025
5 512 234 2051 888 2049	2049
10 1 1 1 1 1 4097 1 1 1 1	4097
8 2456 23445 4086 511 4096 256 23546 854	4086
6 1001 1001 513 1001 1001 1001	513
8 2466 23545 4186 521 4036 456 25547 864	864
6 1101 1011 513 10010 10111 10010	1011
10 12 13 14 15 16 4097 17 18 19 10	4097
10 12 13 14 15 16 4096 17 18 19 10	18

Решение

```
program z83z92v3;
```

```
function good(num:integer):boolean;
```

```
var sum:integer;
```

```
begin
```

```
  sum:=0;
```

```
  while num > 0 do
```

```
  begin
```

```
    sum:=sum+num mod 2;
```

```
    num:=num div 2;
```

```
  end;
```

```
good:=sum mod 2 = 0;
```

```
end;
```

```
var x,m,n,i,max:integer;
```

```
begin
```

```
  read(n);
```

```
  max:=0;
```

```
  for i:=1 to n do
```

```
  begin
```

```
    read(x);
```

```
    if good(x) and (x>max) then
```

```
      max:=x;
```

```
  end;
```

```
  writeln(max);
```

```
end.
```

Задача 5

Вася придумывает пароль для каждой новой учетной записи, которую он заводит на каком-то из своих устройств. Время от времени он изменяет пароль, дописывая к нему новые символы. Предыдущий пароль никогда не будет началом пароля для новой учетной записи Васи. После того, как Вася заводит новую учетную запись, он перестает менять пароль на старой.

Зная все Васиные пароли в хронологическом порядке, напишите программу, которая найдет, какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Формат ввода

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

Формат вывода

Вывести одно целое число – какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Примеры

Входные данные	Выходные данные
3 abd abdc bvd	1

4 bcd bcdf bcdfe abc	2
----------------------------------	---

Тесты

Входные данные	Выходные данные
3 abd abdc bvd	1
4 bcd bcdf bcdfe abc	2
5 a aa aaa aaaa aaaaa	4
7 a b cd df fg fg ab	0
8 a b c d e f fg fgh	2
5 a ab c d e	1
1 asdvbn	0
8 a	

ab c cd cde f fg h	
8 a ab abc abcd cde f fg h	3
8 f fg h a ab abc abcd abcde	4

Решение

```
program z85z94v1_1203;
```

```
var
```

```
n,i,x,len,maxlen:integer;
```

```
s,ps:string;
```

```
begin
```

```
  readln(n);
```

```
  maxlen:=0;
```

```
  len:=0;
```

```
  ps:='';
```

```
  for i:=1 to n do
```

```
  begin
```

```
    readln(s);
```

```
    if (pos(ps,s)=1) and (length(ps)<length(s)) then
```

```
      begin
```

```
        len:=len+1;
```

```
      end
```

```
    else
```

```
      begin
```

```
        if len>maxlen then
```

```
          maxlen:=len;
```

```
        len:=0;
```

```
      end;
```

```

ps:=s;
end;
if len>maxlen then
maxlen:=len;
writeln(maxlen);
end.

```

Ситуационная задача 1.

Миша ведет дневник, отслеживая, сколько шагов в день он проходит пешком. Каждый день он записывает количество шагов. Требуется определить, в какой день во время периода наблюдений он прошел наибольшую дистанцию и прошел ли он столько, сколько планировал, за все время.

На вход программе в первой строке подаются натуральное число N – количество дней, в течение которых велось наблюдение ($N \leq 20$) и натуральное число X – запланированное количество шагов.

Далее в N строках подается на вход по целому положительному числу – x_i : количество шагов, пройденных в i -й день.

Вывести два целых числа – в первой строке вывести номер дня, в который Миша прошел больше всего шагов, во второй строке вывести 1, если он прошел не менее запланированного числа шагов и 0 в обратном случае. Если Вася прошел одинаковое количество шагов в несколько разных дней, вывести наименьший номер дня.

Пример

Ввод	Вывод
3 10000 5100 6500 4200	2 1

Тесты

Ввод	Вывод
1 5000 4900	1 0
2 10 6 7	2 1
3 15 5 3 6	3 0
10 50000 4000 4500 5000 5200 7000 8900 8765 2345	9 1

9023 1298	
3 3 1 1 1	1 1

Решение

```
program z801_1103;
```

```
var
```

```
  n,x,j,k,x,xi,xmax:integer;
```

```
begin
```

```
  readln(n,x);
```

```
  k:=0;
```

```
  readln(xmax);
```

```
  xj:=1;
```

```
  k:=xmax;
```

```
  for j:=2 to n do
```

```
    begin
```

```
      readln(xi);
```

```
      if xi>xmax then
```

```
        begin
```

```
          xj:=j;
```

```
          xmax:=xi;
```

```
        end;
```

```
      k:=k+xi;
```

```
    end;
```

```
  writeln(xj);
```

```
  if k>=x then
```

```
    writeln(1)
```

```
  else
```

```
    writeln(0)
```

```
end.
```

Ситуационная задача 2.

Исследовательский аппарат на поверхности Марса может выполнять команды «Фотографирование», «Пробное бурение», «Взятие образцов грунта», «Анализ атмосферы». Из-за конструктивных особенностей на аппарат наложен ряд ограничений. Нельзя выполнять команду «Анализ атмосферы» после команды «Пробное бурение». Нельзя выполнить команду «Фотографирование» после команды «Взятие образцов грунта». Команду «Взятие образцов грунта» можно выполнять только следующей после команды «Пробное бурение». Никакую команду, кроме команды «Взятие образцов грунта», нельзя выполнить подряд дважды.

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n , если до начала выполнения программы аппарат выполнил команду «Пробное бурение».

На вход программе подается натуральное число n ($n \leq 15$) – количество команд.

Вывести целое число – количество выполнимых последовательностей команд длиной n .

Пример

Ввод	Вывод
2	5

Тесты

Ввод	Вывод
1	2
3	11
5	56
12	16212
15	183922

Решение

```
program z8102_1103;
```

```
var
```

```
  i,n,a,b,c,d,pa,pb,pc,pd:integer;
```

```
begin
```

```
  readln(n);
```

```
  a:=0; //Фото
```

```
  b:=1; //Бурение
```

```
  c:=0; //Грунт
```

```
  d:=0; //Атмосфера
```

```
  for i:=1 to n do
```

```
    begin
```

```
      pa:=a; pb:=b; pc:=c; pd:=d;
```

```
      a:=pb+pd;
```

```
      b:=pa+pc+pd;
```

```
      c:=pb+pc;
```

```
      d:=pa+pc;
```

```
    end;
```

```
  writeln(a+b+c+d);
```

```
end.
```