

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

**10 класс  
Вариант 1**

**Задача 1**

Вася наблюдает, как строители копают фундамент для нового дома. Землю они вывозят одинаковыми самосвалами. По каким-то своим причинам всю выбранную землю вывозят со стройки в тот же день, в который ее выбрали, то есть даже за одним кубометром все равно вечером приедет самосвал. Ещё одна странность, которую заметил Вася, что первый в день самосвал всегда наполняли максимум на половину.

Васе известно, сколько дней копали фундамент, какой объем земли вывозили каждый день и сколько кубометров помещается в самосвал. Вася любил считать всё, что можно, и в этот раз он решил считать самосвалы, но отвлекался на домашнюю работу и теперь боится, что посчитал не верно. Напишите программу, которая поможет Васе проверить общее количество самосвалов, которые вывозили грунт.

В первой строке на вход программы подаются два натуральных числа: количество дней  $t$ , в течение которых копали фундамент,  $1 < t \leq 10^3$  и объем земли  $v$ , который вмещается в один самосвал,  $1 < v \leq 10^2$ . На следующей строке на вход программы поступает  $n$  целых неотрицательных чисел:  $a_1, a_2, \dots, a_n$ .  $a_i$  – сколько земли выбрали в  $i$ -й день,  $1 < a_i \leq 10^5$

Выведите одно целое число – количество требуемых самосвалов.

**Пример**

Входные данные	Выходные данные
2 10 10 15	4

*Комментарий к примеру.* Строители работали два дня, в первый день они извлекли 10 кубометров грунта, и увезли двумя самосвалами по 5 кубов каждый, на следующий день тоже понадобилось 2 самосвала: первый вывез 5, второй 10 кубов. Итого потребовалось 4 самосвала.

**Тесты**

Входные данные	Выходные данные	Баллы
2 10 10 15	4	0
2 10 5 5	2	10
2 10 2 2	2	10
2 5 2 2	2	10
2 5 3 3	4	10
2 10 6 6	4	10

10 4 11 2 3 4 5 6 7 8 9 10	25	10
10 6 11 2 3 4 5 6 7 8 9 10	20	10
10 2 11 2 3 4 5 6 7 8 9 10	40	10
10 3 11 2 3 4 5 6 7 8 9 10	32	10
10 100 11 2 3 4 5 6 7 8 9 10	10	10

### Решение

```
n, v0 = map(int, input().split())
days = list(map(int, input().split()))
res = 0
for v in days:
    res += int(((v + 3 * v0 / 2 - 0.5) // v0) if v > 0 else 0)
print(res)
```

### 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

**СВА** – 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;

```

```

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 asdvpn	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.

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.
```

### Ситуационная задача 2.

Беспилотник летит над большим озером в спокойную погоду. В него заложена программа, в соответствии с которой он изменяет высоту полета один раз в минуту. В программе могут быть следующие команды:

«Выше» – беспилотник поднимается на 0,5 м.

«Ниже» – беспилотник опускается на 0,5 м.

«Сохранение высоты» - беспилотник сохраняет высоту до следующей команды. Таким образом, все перемещения беспилотника происходят с шагом 0.5 м. Набор высоты или снижение происходят за время, пренебрежимо малое по сравнению с минутой. Если беспилотник касается воды, он выходит из строя. Зная начальную высоту и продолжительность полета, найдите,

сколькими способами можно задать последовательность команд, не приводящую к разрушению беспилотника, после выполнения которой он окажется в заданном диапазоне высот.

На вход программе подаются вещественное число  $h_0$  ( $1 \leq h_0 \leq 100$ ) – начальная высота квадрокоптера, целое число  $t$  ( $t \leq 20$ ) – продолжительность полета, два вещественных числа  $l$  и  $h$  ( $1 \leq l \leq h \leq 100$ ) – нижняя и верхняя граница требуемого диапазона высот. Все вещественные числа даны с точностью до 0,5 м.

Программа должна вывести одно целое число – количество способов.

### Пример

Ввод	Вывод
2 2 2.5 3	3

### Тесты

Ввод	Вывод
2 2 2.5 3	3
1 10 1 10	22808
1 15 0.5 0.5	542895
10 20 1 2	10185
10 10 1 10	34001
5 10 1 10	59038

### Решение

```

program z10var1;
const
  TMAX=20;
  HMAX=220;
type
  tmatrix = array [0..TMAX,1..HMAX] of integer;

procedure init(var matrix:tmatrix);
var i,j:integer;
begin
  for i:=1 to TMAX do
    For j:=1 to HMAX do
      begin
        matrix[i,j]:=0;
      end;
end;

procedure work(var matrix:tmatrix; h0,t,l,h:integer);
var i,j,sum:integer;
begin
  matrix[0,h0]:=1;
  for i:=1 to t do
    begin
      for j:=1 to HMAX do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j];
    end;
end;

```

```

for j:=1 to HMAX-1 do
  matrix[i,j]:=matrix[i,j]+matrix[i-1,j+1];
for j:=2 to HMAX do
  matrix[i,j]:=matrix[i,j]+matrix[i-1,j-1];
end;
sum:=0;
for i:=1 to h do
  sum:=sum+matrix[t,i];
  writeln(sum);
end;

var
  matrix:tmatrix;
  h0,t,l,h:integer;
  h0r,lr,hr:real;
begin
  init(matrix);
  readln(h0r,t,lr,hr);
  h0:=round(2*h0r);
  l:=round(2*lr);
  h:=round(2*hr);
  work(matrix,h0,t,l,h);
end.

```