

Олимпиада школьников «Шаг в будущее»
Отборочный этап

10 класс

Вариант 1

Задача 1

Для совершения обгона требуется, чтобы видимость дороги была не менее 250 метров. Водитель маршрутки Ахмет хочет обогнать рейсовый автобус, который, по его мнению, движется слишком медленно. До ближайшего столба впереди по движению осталось y метров. Сколько столбов освещения должен видеть Ахмет, чтобы видимости дороги для обгона было гарантировано достаточно? Расстояние между столбами освещения составляет z метров.

На вход программы в одной строке через пробел подаются 2 целых числа: неотрицательное число y и натуральное число z ($z > y$), каждое из которых не превышает 10^9 .

В результате работы программа должна вывести одно целое число – минимальное количество столбов, гарантирующих видимость 250 метров дороги.

Пример

Входные данные	Выходные данные
10 50	6

Комментарий к примеру

Первый столб освещения находится на расстоянии 10 метров от маршрутки, 5-ый – 210 метров, 6-ой – 260 метров. Только 6-ой столб даёт гарантию видимости 250 метров дороги.

Тесты

Входные данные	Выходные данные
10 50	6
1 50	6
0 50	6
249 250	2
250 251	1
250 1000000	1
0 251	1
0 250	2
0 1	251
249 1000000	2

Решение

```
x = 250
y, z = map(int, input().split())
d, n = y, 1
while d < x:
    d += z
    n += 1
print(n)
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

Задача 2

На вход программе подаются натуральные числа. Точное количество чисел неизвестно, но может быть очень велико. Требуется найти среди введенных данных наибольшее число, двоичная запись которого оканчивается двумя нулями. Гарантируется, что хотя бы одно такое число в последовательности есть.

Формат ввода

В строке вводится сначала целое число n – количество чисел в цепочке ($n \leq 1000$), затем n натуральных чисел (каждое число $\leq 10^6$), все числа отделены друг от друга одним или несколькими пробелами

Формат вывода

Вывести одно целое число – наибольшее число, двоичная запись которого оканчивается двумя нулями.

Пример

Входные данные	Выходные данные
5 30 1000 35 6 39	1000

Тесты

Входные данные	Выходные данные
5 30 1000 35 6 39	1000
7 80 33 152 67 159 76 68	152
4 15 48 52 87	52
5 15 24 34 13 102	24
8 18 8 16 24 48 92 102 88	92

Решение

```
program z83z92v1;

function good(num:integer):boolean;
begin
    good:=num mod 4 = 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;
    end;
end;
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

```
end;  
writeln(max);  
end.
```

Задача 3

Назовем цифровым корнем натурального числа X по основанию 5 цифру, полученную в результате работы следующего алгоритма:

складываем все цифры пятеричного представления числа X в системе счисления $p = 5$, у полученной суммы вновь складываем цифры таким же образом и так далее до тех пор, пока не останется одна цифра. Это и есть цифровой корень.

Например, $X = 2977_{10} = 43402_5$. Сумма цифр числа X есть 23_5 . Складываем еще раз 10_5 . Еще одна итерация и мы знаем цифровой корень числа – это 1.

Задается интервал $[a, b]$. Надо определить, какая цифра является цифровым корнем по основанию 5 для большего количества чисел на этом отрезке. Если таких цифр несколько, то выведите наименьшую.

На ввод подаются два десятичных натуральных числа a и b . Оба числа не превосходят 10^6 .

На выходе одно число – цифра – решение задачи.

Пример

Входные данные	Выходные данные	Примечание
2976 2978	1	Число Цифровой корень 2976 4 2977 1 2978 2 Каждая из цифр 1, 2 и 4 является цифровым корнем для одного числа из интервала. Наименьшая цифра – 1
2976 2984	4	Цифра 4 является корнем для трех чисел на интервале 2976, 2980 и 2984. Остальные цифры для меньшего количества чисел являются корнями.

Тесты

Входные данные	Выходные данные
1 10	1
6 10	2
4 24	4

Олимпиада школьников «Шаг в будущее»
Отборочный этап

111 131	3
1 1000000	1
999998 1000000	2
999990 999997	1
999991 999999	3
99999 999999	3
11111 99999	3

Решение

$p = 5$

```
def f(n):
    while n >= p:
        s = 0
        while n > 0:
            s += n % p
            n = n // p
        n = s
    return n

digit = [0]*5
a, b = map(int, input().split())
for x in range(a, b + 1):
    digit[f(x)] += 1
mx = max(digit)
print(digit.index(mx))
```

Задача 4

Определите абсолютную погрешность измерения физической величины по методу Корнфельда: половина разности между максимальным и минимальным измерением.

Формат ввода

В первой строке программы вводится натуральное число N – количество измерений. $2 \leq N \leq 300$. Далее в N строчках вводится по одному целому числу x_i ($-1000 \leq x_i \leq 1000$) – результат измерения под номером i .

Формат вывода

Вывести через пробел одно целое число – целую часть абсолютной погрешности, определенной по методу Корнфельда.

Пример

Входные данные	Выходные данные
5 3 2	3

Олимпиада школьников «Шаг в будущее»
Отборочный этап

8	
7	
6	

Тесты

Входные данные	Выходные данные
5 3 2 8 7 6	3
4 1 8 2 3	3
4 2 2 2 2	0
8 -7 5 2 3 1 -1 0 -1	6
5 -7 -2 -3 -1 -6	3

Решение

```
program pzv1;  
  
var  
  i, x, n, min, max: integer;  
begin  
  readln(n);  
  min:=1001;
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

```
max:=-1001;  
for i:=1 to n do  
begin  
  readln(x);  
  if x>max then  
  begin  
    max:=x;  
  end;  
  if x<min then  
  begin  
    min:=x;  
  end;  
end;  
writeln(trunc((max-min)/2));  
end.
```

Задача 5

Несколько датчиков сообщают о поведении смеси внутри лабораторного реактора, передавая кодовые сообщения, состоящие из букв и цифр. Из-за физико-химических свойств смеси некоторые датчики могут передавать данные с ошибками. Если сообщение не содержит ошибок, то восьмеричная запись суммы цифр в этом сообщении заканчивается на 6. Найдите количество сообщений, не содержащих ошибок.

Формат ввода

В первой строке вводится сначала натуральное число n ($n \leq 1000$) – количество датчиков, затем в n следующих строках записано по одному сообщению. Сообщение состоит из строчных латинских букв и цифр, его длина не превышает 30 символов.

Формат вывода

Вывести одно целое число – количество сообщений, не содержащих ошибок.

Примеры

Входные данные	Выходные данные
3 a6b abcdfresd11211f asdv7	2

Тесты

Входные данные	Выходные данные
3 a6b abcdfresd11211f asdv7	2

Олимпиада школьников «Шаг в будущее»
Отборочный этап

4 dasdfs asddf2 awddewf0 adf3212	0
4 adb60 dfs213 ds3 sadf2345	3
6 afd6 a1d2s3 f2d a4sdr df55g d0f3r2	2
3 sad1 as1 sda	0

Решение

```
program szv1;
const
  digits = '1234567890';

function count(s:string):integer;
var
  i,k:integer;
begin
  k:=0;
  for i:=1 to length(s) do
    begin
      k:=k+(pos(copy(s,i,1),digits) mod 10);
    end;
  count:=k;
end;

var
  n,i,c:integer;
  s:string;
begin
  readln(n);
  c:=0;
  for i:=1 to n do
    begin
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

```
readln(s);  
if (count(s) mod 8) = 6 then  
    c:=c+1;  
end;  
writeln(c);  
end.
```

Задача 6 - Ситуационная задача

Для работы манипулятора необходимо собрать электрическую цепь, последовательно соединяя резисторы. В наличии несколько резисторов. Зная их сопротивления, определите, сколькими способами можно собрать цепь. Использовать все резисторы не обязательно.

Формат ввода

В первой строке программы вводится натуральное число N ($N \leq 20$) – количество резисторов. Во второй строке указывается натуральное число R ($R \leq 20000$) – требуемое сопротивление цепи. Далее в N строчках вводится по одному натуральному числу r_i ($r_i \leq 2000$) – сопротивление резистора под номером i . Все сопротивления указываются в килоомах.

Формат вывода

Вывести одно целое число – количество способов, которыми можно собрать цепь.

Пример

Входные данные	Выходные данные
4 5 1 4 3 2	2

Тесты

Входные данные	Выходные данные
4 5 1 4 3 2	2
5 5 1 1 1 1	4

Олимпиада школьников «Шаг в будущее»
Отборочный этап

2	
5 6 1 1 1 1 2	1
8 15 12 3 4 8 11 10 5 8	5
8 10 1 2 3 1 2 4 1 5	24

Решение

```
program z2_3v;
const
  MAXN = 20;

type
  tresist = array [1..MAXN] of integer;

procedure init(var used:tresist);
var
  i:integer;
begin
  for i:=1 to MAXN do
    used[i]:=0;
  end;

procedure fill(var resist:tresist; var n:integer);
var
  i:integer;
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

```
begin
  for i:=1 to n do
    begin
      readln(resist[i]);
    end;
  end;

function shift(var used:tresist; n:integer):boolean;
var
  i:integer;
  over:boolean;
begin
  i:=n;
  over:=true;
  while over and (i>=1) do
    begin
      used[i]:=used[i]+1;
      if used[i]>1 then
        begin
          used[i]:=0;
          i:=i-1;
        end
      else
        over:=false;
      end;
    shift:=over;
  end;

function calc(used, resist:tresist; n,r:integer):boolean;
var
  i:integer;
  sm:integer;
begin
  for i:=1 to n do
    begin
      if used[i]=1 then
        begin
          sm:=sm+resist[i];
        end;
      end;
    calc:=(sm=r);
  end;

function work(used, resist:tresist; n,r:integer):integer;
var
  over:boolean;
  f:integer;
begin
  over:=false;
  while not over do
    begin
      if calc(used, resist, n, r) then
        f:=f+1;
      end;
    end;
  end;
```

Олимпиада школьников «Шаг в будущее»
Отборочный этап

```
    over:=shift(used,n);  
end;  
work:=f;  
end;  
  
var  
    used,resist:tresist;  
    n,r:integer;  
begin  
    init(used);  
    readln(n,r);  
    fill(resist,n);  
    writeln(work(used,resist,n,r));  
end.
```