Олимпиада школьников «Шаг в будущее»
Заключительный этап

10 класс
Вариант 1

Задача 1

Крупную плитку кладут сверху вниз, начиная со среднего вертикального ряда, чтобы положенная плитка имела осью симметрии вертикаль по середине стены. Если на оставшееся справа или слева место плитка не помещается целиком, её режут и кладут резаной стороной к краю стены. Тоже касается низа стены: если внизу плитка не укладывается по всей высоте, её режут по горизонтали и кладут резаной стороной вниз. Резаной стороной плитка может упираться только в правый, левый или нижний край стены. В данной задаче рассмотрим плитку, рисунок которой позволяет её класть, будучи перевёрнутой на 180 градусов. На самом деле важны ещё расстояния между соседними плитками, но в данной задаче будем считать, что плитка кладётся впритык. Вам не нужно рассчитывать полный расход, рассчитайте сколько плиток необходимо на один верхний ряд.

На вход программы в одной строке подаются 2 натуральных числа: ширина одной плитки, ширина стены. Все числа не превышают 10^6. Ширины стены и плитки измеряются чётным количеством миллиметров. Одна плитка точно может быть положена на стену, не будучи разрезанной.

В результате работы программа должна вывести одно целое число: ответ задачи.

Пример

Входные данные	Выходные данные
500 1010	3

Комментарий к примеру

Положив центральную плитку, слева и справа останется суммарно 510 мм, из них 255 мм слева от центральной вертикали и столько же справа. Поэтому слева и справа придётся расходовать ещё по целой плитке, отрезав от каждой по 245 мм. Итого на один горизонтальный ряд будет уходить 3 плитки.

Тесты

Входные данные	Выходные данные
500 1010	3
500 1000	2
500 1002	3
498 1000	3
2 200000	100000

Решение

```
wp, ws = map(int, input().split())
side_width = (ws - wp) // 2
n = side_width // wp * 2 + 1 # количество целых плиток на ряд
side_left = side_width % wp
if side_left > 0: # если для укладки справа и слева придётся
резать
    # если можно использовать одну плитку на обе стороны
    if side_left <= wp // 2:
        n += 1
    else:
        n += 2
print(n)
```

Задача 2

Для того, чтобы открыть электронный замок, суперагентам Саше и Жене необходимо получить цифровой ключ. Код можно получить следующим образом. Необходимо взять все девятеричные числа длиной ***N***. Подсчитать, сколько среди них таких чисел, у которых в троичной записи нет ни одного значащего нуля. Полученный результат и будет цифровым ключом к замку. Напишите программу, которая поможет Саше и Жене получить код замка.

Входные данные:

На вход программе подаётся одно целое число ***N (0<N<30)***.

Выходные данные:

Вывести одно целое число – цифровой код замка.

Пример:

Входные данные	Выходные данные
2	***24***

Тесты

Входные данные	Выходные данные
1	6
2	24
3	96
4	384
5	1536
8	98304
10	1572864
15	1610612736
20	1649267441664
29	432345564227567616

```
var
  c:int64;
  n:integer;
begin
  readln(n);
  c:=6;
  for var i:=1 to n-1 do
    c:=c*4 ;
  writeln(c)
end.
```

Задача 3

Разведывательный космический корабль “Пионер космоса” наткнулся на новую цивилизацию на планете, которую назвали Крампо. Контакт с аборигенами установлен. Выяснилось, что у них принята 6-ричная система счисления. Причем запись позиционная, а иероглифы цифр похожи на наши скобки.

Во время длительного ночного дежурства связист решил проверить, сколько чисел, записанных на языке Крампо, являются правильными скобочными последовательностями (ПСП). Помогите ему. Напишите программу, которая считает, сколько чисел в заданном диапазоне будут выглядеть на Крампо для землянина как правильные скобочные последовательности.

Таблица цифр Крампо:

0	1	2	3	4	5
<	>	[	]	{	}

Правильная скобочная последовательность определяется следующим образом:
Пустая последовательность является правильной.
Если A – правильная, то последовательности <A>, [A], {A} – правильные.
Если A и B – правильные последовательности, то последовательность AB – правильная.

Входные данные

В строке задаются два натуральных числа 0 < X <= Y <= 1 000 000.

Выходные данные

Выводится количество целых чисел на отрезке [X; Y], являющихся правильными скобочными последовательностями в записи на языке Крампо.

Вход	Выход	Примечание
700228 700231	1	$700228_{10} = 23001444_6$ = []<<>{{{ не ПСП: 4 скобки не закрыты
		$700229_{10} = 23001445_6$ = []<<>{{} не ПСП: 2 скобки не закрыты
		$700230_{10} = 23001450_6$ = []<<>{}< не ПСП: 2 скобки не закрыты
		$700231_{10} = 23001451_6$ = []<<>{}> **ПСП**
34 37	0	$34_{10} = 54_6$ = }{ не ПСП: открывающая скобка раньше закрывающей
		$35_{10} = 55_6$ = }} не ПСП: закрывающих больше, чем открывающих

		$36_{10} = 100_6 =$ >< не ПСП: открывающая скобка раньше закрывающей
		$37_{10} = 101_6 =$ ><> не ПСП: закрывающая скобка без открывающей

Тесты

№	Ввод	Вывод
1	1 100	2
2	700228 701000	3
3	1 1000000	482
4	700891 700900	1
5	700000 800000	215
6	800000 900000	0
7	1 10	0
8	1 50	2
9	111 1111	12
10	1073 1073	1

Решение

```
def krampo(x):  # перевод числа на Крампо
  sk = '<>[]{}'
  ans = ''
  while x:
    c = x % 6
    x = x // 6
    ans = sk[c] + ans
  return ans


def pcp(a): # проверка на ПСП
  sk1 = '{[<'
```

```
  sk2 = '}]>'
  stack = []
  for char in a:
    if char in sk1:
      stack.append(char)
    elif char in sk2:
      if not stack:
          return False
      else:
        top = stack.pop()
        if sk1[sk2.index(char)] != top:
          return False
  else:
      return True if not stack else False

x, y = map(int, input().split())
cnt = 0
for num in range(x, y + 1):
  if pcp(krampo(num)):
    cnt += 1
print(cnt)
```

Задача 4

Школьник Вася играет в шахматы на онлайн-портале. После каждого турнира его рейтинг пересчитывается в зависимости от результатов. Так как Вася интенсивно и старательно учится, его успехи в шахматах очень непостоянны; как только он налегает на учебу, он начинает играть хуже, а его рейтинг начинает падать. Когда рейтинг падает слишком сильно (по субъективному мнению Васи), Вася начинает интенсивно тренироваться. Когда рейтинг поднимается достаточно высоко (тоже по его субъективному мнению), он оставляет тренировки. Известно, что субъективное мнение Васи – величина переменная. Также известно, что когда он интенсивно тренируется, его рейтинг обязательно растет, а когда он перестает тренироваться, он сразу начинает падать. Зная, как менялся рейтинг Васи после каждого еженедельного турнира, определите, сколько недель длился самый долгий цикл интенсивных тренировок. В начале сезона Вася не тренировался.

Формат ввода

В первой строке программы вводится натуральное число **N** – количество измерений. $2 \le \mathbf{N} \le 50$. Далее в **N** строчках вводится по одному целому числу $\mathbf{x_i}$ ($0 \le \mathbf{x_i} \le 2000$) – рейтинг Васи после турнира на неделе номер **i**.

Формат вывода

Вывести через пробел одно целое число – сколько недель длился самый долгий цикл интенсивных тренировок.

Пример

Входные данные	Выходные данные
4 998 997	2

999 1001	

Тесты

Входные данные	Выходные данные
5 997 1003 995 999 1000 1001 1005 1000	4
6 1001 998 1000 996 1000 1001	2
2 0 1	1
5 100 99 98 97 96	0
5 1003 1001 1000 999 1002	1

Решение

```
program pzv1;

var
 i,px,x,n,len,maxlen:integer;
 flag:boolean;
begin
 readln(n);
 maxlen:=0;
```

```
 len:=0;
 flag:=false;
 readln(x);
 for i:=2 to n do
 begin
   px:=x;
   readln(x);
   if x>px then
   begin
    if flag then
    begin
      len:=len+1;
    end
    else
    begin
      flag:=true;
      len:=1;
    end;
   end;
   if x<px then
   begin
    if flag then
    begin
      if len>maxlen then
        maxlen:=len;
      len:=0;
      flag:=false;
    end
   end;
 end;
 if len>maxlen then
  maxlen:=len;
 writeln(maxlen);
end.
```

Задача 5.

Система датчиков отслеживает химические процессы внутри реактора, передавая цифровые коды в восьмеричной системе счисления. Некоторые датчики могут передавать данные с ошибками. Если сообщение не содержит ошибок, то шестнадцатеричная запись суммы цифр в этом сообщении заканчивается на **A**. Найдите количество сообщений, не содержащих ошибок.

Формат ввода

В первой строке вводится сначала натуральное число n ($n \le 1000$) – количество датчиков, затем в n следующих строках записано по одному сообщению. Сообщение состоит из цифр от 0 до 7, его длина не превышает 30 символов.

Формат вывода

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

Примеры

Входные данные	Выходные данные
4 7676462 746764 77336 31337	2

Тесты

Входные данные	Выходные данные
4 7676462 746764 77336 31337	1
4 31337 746764 77336 7676466	2
6 767 77255 43 77255 77255 77255	4
2 33 12	0
1 7775	1

Решение

```
program szv1;
const
 digits = '123456789abcdef';


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);
  end;
```

```
  count:=k;
end;

var
 n,i,c:integer;
 s:string;
begin
 readln(n);
 c:=0;
 for i:=1 to n do
   begin
     readln(s);
     if (count(s) mod 10) = 5 then
       c:=c+1;
     end;
   writeln(c);
end.
```

Задача 6 - Ситуационная задача

Для того, чтобы, нужно обеспечить напряжение, последовательно включив в цепь несколько источников. В наличии есть несколько источников напряжения. Каждый из них может работать в нормальном или форсированном режиме. В форсированном режиме напряжение источника увеличивается в два раза. Так как в форсированном режиме источник быстрее выйдет из строя, форсировать источники нежелательно. Зная напряжение для каждого источника, определите, какое минимальное количество источников потребуется, чтобы обеспечить требуемое напряжение, и сколько из них будет форсировано. Если существуют две комбинации с одинаковым количеством задействованных источников, следует вывести ту, где форсировано меньше источников.

Формат ввода

В первой строке программы вводится натуральное число **N** (**N** ≤ 20) – количество источников. Во второй строке указывается целое число **U** (0 ≤ **U** ≤ 2000) – требуемое напряжение. Далее в **N** строчках вводится по одному натуральному числу $\mathbf{u_i}$ ($\mathbf{u_i}$ ≤ 380) – штатное напряжение источника под номером **i**. Все напряжения указываются в вольтах.

Формат вывода

Вывести два целых числа – минимально необходимое количество источников, необходимых для набора напряжения и количество форсированных источников. Напряжение, обеспечиваемое источниками, должно строго совпадать с требуемым. Если обеспечить такое напряжение невозможно, вывести *0 0*.

Пример

Входные данные	Выходные данные
4 42 20 2 11 10	2 1

Тесты

Входные данные	Выходные данные
4 42 20 2 11 10	2 1
3 20 9 11 20	1 0
3 20 9 11 10	1 1
5 10 1 2 3 4 5	1 1
1 20 7	0 0

Решение

```
program sz1010_1;
const
 N = 20;
type
 tmas = array [1..N] of integer;

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

procedure fill(var mas:tmas; var k:integer);
var
 i:integer;
begin
 for i:=1 to k do
 begin
```

```
    readln(mas[i]);
  end;
end;

procedure rebuild(var used:tmas; num:integer);
var
  i:integer;
begin
  i:=1;
  while num > 0 do
  begin
    used[i]:=num mod 3;
    num:=num div 3;
    i:=i+1;
  end;
end;

procedure check(mas,used:tmas; u,num,kol:integer; var
min,force:integer);
var
  i,sum,mi,fo:integer;
begin
  sum:=0;
  rebuild(used,num);
  mi:=0;
  fo:=0;
  for i:=1 to kol do
  begin
    if used[i] = 1 then
    begin
      mi:=mi+1;
      sum:=sum+mas[i];
    end;
    if used[i] = 2 then
    begin
      mi:=mi+1;
      fo:=fo+1;
      sum:=sum+2*mas[i];
    end;
  end;
  if sum = u then
  begin
    if (mi<min) or ((mi = min) and (force = fo)) then
    begin
      min:=mi;
      force:=fo;
    end;
  end;
end;

var
 kol,i,min,force,border,u:integer;
 used,mas:tmas;
```

```
begin
 readln(kol);
 readln(u);
 min:=N+1;
 force:=N+1;
 init(used);
 fill(mas,kol);
 border:=round(power(3,kol));
 for i:=1 to border do
   check(mas,used,u,i,kol,min,force);
 if (min = N+1) and (force = N+1) then
 begin
   min:=0;
   force:=0;
 end;
 write(min,' ',force);
end.
```