

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**8 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)
```

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**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” наткнулся на новую цивилизацию на планете, которую назвали Крампо. Контакт с аборигенами установлен.

# Олимпиада школьников «Шаг в будущее»

## Заключительный этап

Выяснилось, что у них принята двоичная 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
)	(

**Правильная скобочная последовательность** определяется следующим образом:

Пустая последовательность является правильной.

Если A – правильная, то последовательности (A) – правильная.

Если A и B – правильные последовательности, то последовательность AB – правильная.

### Входные данные

В строке задаются два натуральных числа  $0 < X \leq Y \leq 1\,000\,000$ .

### Выходные данные

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

Вход	Выход	Примечание
177 181	2	$177_{10} = 10110001_2 = ()(())($ не ПСП: открывающая скобка раньше закрывающей
		$178_{10} = 10110010_2 = ()(())() $ ПСП
		$179_{10} = 10110011_2 = ()(())($ не ПСП: 2 скобки не закрыты
		$180_{10} = 10110100_2 = ()(())() $ ПСП
		$181_{10} = 10110101_2 = ()(())($ не ПСП: 2 скобки не закрыты
3 6	0	$3_{10} = 11_2 = (($ не ПСП: 2 скобки не закрыты
		$4_{10} = 100_2 = ())$ не ПСП: закрывающих скобок больше, чем открывающих
		$5_{10} = 101_2 = ()($ не ПСП: открывающая скобка без закрывающей

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

		$6_{10} = 110_2 = (())$ не ПСП: открывающая скобка без закрывающей
--	--	--

Тесты

№	Ввод	Вывод
1	1 100	8
2	700228 701000	10
3	1 1000000	21074
4	700891 700900	0
5	700000 800000	4768
6	800000 900000	3826
7	1 10	2
8	30 50	3
9	728 730	1
10	13016 13016	1

Решение

```
def ok(n): # разбор числа на цифры и проверка на ПСП
    k = 0
    while n > 0:
        if n % 2 == 0:
            k += 1
        else:
            k -= 1
        if k < 0:
            return False
        n //= 2
    return k == 0

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

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

```
if ok(num):  
    cnt += 1  
print(cnt)
```

#### Задача 4

Школьник Коля играет в шахматы на онлайн-портале. После каждой игры его рейтинг пересчитывается в зависимости от результатов. Зная, как менялся рейтинг Коли после каждого еженедельного турнира, определите, насколько его наилучший рейтинг превышает минимум.

*Формат ввода*

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

*Формат вывода*

Вывести через пробел одно целое число – разницу между его наилучшим и минимальным рейтингом.

#### Пример

Входные данные	Выходные данные
4 998 997 999 1001	4

#### Тесты

Входные данные	Выходные данные
5 997 1003 995 1005 1000	5
6 1001 998 1000 996 1000 1000	5
2 0 0	1
5 100	100

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

3 100 100 0	
5 1001 1000 1000 999 1002	3

**Решение**

```
program pzv1;  
  
var  
  i,x,n,mid,max:integer;  
begin  
  readln(n);  
  mid:=0;  
  max:=-1;  
  for i:=1 to n do  
    begin  
      readln(x);  
      mid:=mid+x;  
      if x>max then  
        begin  
          max:=x;  
        end;  
    end;  
  mid:=mid div n;  
  writeln(max-mid);  
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  $n$  ( $n \leq 1000$ ) – 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.

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**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;  
end;
```

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

```
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.  
end.
```

### Задача 6 - Ситуационная задача

Для того, чтобы нагреть воду в колбе, используют таблетки с топливом разных типов. Можно выбрать только один тип таблеток. Зная удельную теплоту сгорания, массу топлива и цену каждого типа таблетки, определите, какой тип таблеток выгоднее всего использовать. КПД нагревателя принять за 30%. Удельная теплоемкость воды – 4200 Дж/кг·К. Воду нагревают на 20 К. Если таблетка поджигается, потушить ее уже нельзя.

*Формат ввода*

В первой строке вводится натуральное число  $N$  – количество таблеток.  $1 \leq N \leq 15$ . Во второй строке вводится вещественное число  $M$  – масса воды, которую надо вскипятить. Далее в  $N$  строчках вводится по три вещественных числа  $m_i$ ,  $q_i$  и  $p_i$  – масса топлива и удельная теплота сгорания у таблетки под номером  $i$ . Все величины указываются в системе СИ, положительные, не превосходят 10000 и имеют физический смысл. Во входных данных нет двух таблеток с одинаковыми характеристиками.

*Формат вывода*

Вывести через пробел одно целое число – номер типа таблеток, которые выгоднее всего использовать.

### Пример

Входные данные	Выходные данные
3 3 0.1 3.2 10 0.05 4.5 11 0.08 4.4 9	3

### Тесты

Входные данные	Выходные данные
3 3 0.1 3.2 10 0.05 4.5 11	3



Олимпиада школьников «Шаг в будущее»  
Заключительный этап

0.08 4.4 9	
1 1 0.8 1.2 1	1
2 0.05 1 6500 2.1 1 5000 2	2
5 20 30 5 10 20 6 11 30 7 12 20 5 11 14 5 8	3
2 0.05 1 5000 1.9 1 6200 2.0	1

**Решение**

```

program sz89_1;
const
  C = 4200;
  DT = 20;
  KPD = 0.3;
var
  n,i,nopt,kol:integer;
  q,m,mi,pi,qi,min,z:real;
begin
  readln(n);
  readln(m);
  q := C*DT*m/KPD;
  min:=10000*10000+1;
  nopt:=0;
  for i:=1 to n do
    begin
      readln(mi,qi,pi);
      z:=q/mi/qi;
      kol:=trunc(z);
      if frac(z)>0 then
        kol:=kol+1;
      if pi*kol <min then
        begin
          nopt:=i;
          min := pi*kol;
        end;
      end;
  writeln(nopt);
end.

```