

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**10 класс**  
**Вариант 2**

**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. Рассчитайте сколько отходов будет от одного верхнего ряда уложенной плитки при минимальном количестве израсходованных целых плиток.

На вход программы в одной строке подаются 3 натуральных числа: ширина и высота одной плитки, ширина стены. Все числа не превышают  $10^6$ . Ширина стены и размеры плитки измеряются чётным количеством миллиметров. Одна плитка точно может быть положена на стену, не будучи разрезанной.

В результате работы программы должна вывести одно целое число: площадь в  $\text{мм}^2$  плитки, которая будет отрезана и уйдёт в отходы с одного верхнего горизонтального ряда.

**Пример**

Входные данные	Выходные данные
500 250 1010	122500

**Комментарий к примеру**

Положив центральную вертикаль, 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, из них у обеих крайних не использовано будет  $(245 \times 250) * 2 \text{ мм}^2 = 122500 \text{ мм}^2$ .

**Тесты**

Входные данные	Выходные данные
500 250 1010	122500
500 250 1000	0
500 250 1002	124500
498 2 1000	988
502 2 1000	8

**Решение 1**

```
wp, hp, ws = map(int, input().split())
side_width = (ws - wp) // 2
n = side_width // wp * 2 + 1 # количество целых плиток на ряд
waste = 0 # пока обходимся без отходов
side_left = side_width % wp
```

## Олимпиада школьников «Шаг в будущее»

### Заключительный этап

```
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
        waste += hp * (wp - 2 * side_left)
    else:
        n += 2
        waste += 2 * hp * (wp - side_left)
print(waste)
```

Решение 2

```
wp, hp, 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 * wp * hp - ws * hp)
```

### Задача 2

Для того, чтобы открыть электронный замок, суперагентам Паше и Маше необходимо получить цифровой ключ. Код можно получить следующим образом. Необходимо взять все числа в двадцатипятеричной системе счисления длиной  $N$ . Подсчитать, сколько среди них таких чисел, у которых в пятеричной записи нет ни одной нечетной цифры. Полученный результат и будет цифровым ключом к замку. Напишите программу, которая поможет Паше и Маше получить код замка.

#### *Входные данные:*

На вход программе подаётся одно целое число  $N$  ( $0 < N < 20$ ).

#### *Выходные данные:*

Вывести одно целое число – цифровой код замка.

#### *Пример:*

Входные данные	Выходные данные
2	72

### Тесты

Входные данные	Выходные данные
1	8
2	72
3	648
4	5832
5	52488
6	472392
8	38263752
10	3099363912
15	183014339639688
19	1200757082375992968

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

Решение

```
var  
  c:int64;  
  n:integer;  
begin  
  readln(n);  
  c:=8;  
  for var i:=1 to n-1 do  
    c:=c*9 ;  
  writeln(c)  
end.
```

**Задача 3**

Разведывательный космический корабль “Звёздный странник” наткнулся на новую цивилизацию на планете, которую назвали Хаалт. 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, какие числа, записанные на языке Хаалт, являются правильными скобочными последовательностями (ПСП). Помогите ему. Напишите программу, которая находит в заданном диапазоне **наибольшее** число, которое будет выглядеть на Хаалт для землянина как правильная скобочная последовательность.

*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  $0 < X \leq Y \leq 1\,000\,000$ .

**Выходные данные**

Выводится **наибольшее** целое десятичное число из отрезка  $[X;Y]$ , которое является правильной скобочной последовательностью в записи на языке Хаалт.

Если таких чисел в диапазоне нет, то выводится **0**.

Вход	Выход	Примечание
700228 700231	70023	$700228_{10} = 23001444_6 = []<>\{\}$ не ПСП: 4 скобки не закрыты

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

	1	$700229_{10} = 23001445_6 = []<>\{ \} \text{ не ПСП: 2 скобки не закрыты}$
		$700230_{10} = 23001450_6 = []<>\{ \}< \text{не ПСП: 2 скобки не закрыты}$
		$700231_{10} = 23001451_6 = []<>\{ \}> \text{ ПСП}$
34 37	0	$34_{10} = 54_6 = \} \{ \text{ не ПСП: открывающая скобка раньше закрывающей}$
		$35_{10} = 55_6 = \} \} \text{ не ПСП: закрывающих больше, чем открывающих}$
		$36_{10} = 100_6 = ><< \text{не ПСП: открывающая скобка раньше закрывающей}$
		$37_{10} = 101_6 = ><> \text{не ПСП: закрывающая скобка без открывающей}$

Тесты

№	Ввод	Выход
1	1 100	29
2	700228 701000	700891
3	1 1000000	791817
4	700891 700900	700891
5	700000 800000	791817
6	800000 900000	0
7	1 10	0
8	1 50	29
9	111 1111	1073
10	1073 1073	1073

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**Решение**

```
def ok(n):
    st = []
    while n > 0:
        d = '<>[]{}'[n % 6]
        n //= 6

        if d in '>]}'':
            st.append(d)
        elif len(st) == 0:
            return False
        else:
            d_ = st.pop()
            if (d + d_) not in ['{}', '[]', '<>']:
                return False
    return len(st) == 0

a, b = map(int, input().split())
for n in range(b, a - 1, -1):
    if ok(n):
        print(n)
        break
else:
    print(0)
```

**Задача 4**

Школьник Вася играет в шахматы на онлайн-портале и запоминает победы над сильнейшими соперниками. Каждый сезон он запоминает три наиболее достойные победы. Зная рейтинг оппонентов, с которыми Вася играл в сезоне, определите рейтинги его трех самых сильных оппонентов.

*Формат ввода*

В первой строке программы вводится натуральное число  $N$  – количество измерений.  $3 \leq N \leq 50$ . Далее в  $N$  строчках вводится по одному целому числу  $x_i$  ( $0 \leq x_i \leq 2000$ ) – рейтинг оппонента Васи в игре номер  $i$ .

*Формат вывода*

Вывести через пробел три целых числа в порядке невозрастания – рейтинги самых сильных оппонентов Васи.

**Пример**

Входные данные	Выходные данные
5 11 13 17 19 16	19 17 16

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

Тесты

Входные данные	Выходные данные
5 11 13 17 19 16	19 17 16
6 1 2 3 6 3 4	6 4 3
5 6 6 6 6 6	6 6 6
6 10 15 17 18 18 19	19 18 18
5 1 2 3 4 5	5 4 3

Решение

```
program pzv1;

var
  i,x,n,max1,max2,max3:integer;
begin
  readln(n);
  max1:=-1;
  max1:=-1;
  max1:=-1;
  for i:=1 to n do
    begin
```

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

```
readln(x);
if x > max1 then
begin
  max3:=max2;
  max2:=max1;
  max1:=x;
end
else if x> max2 then
begin
  max3:=max2;
  max2:=x;
end
else if x> max3 then
  max3:=x;
end;
writeln(max1,' ',max2,' ',max3);
end.
```

### Задача 5

Система датчиков отслеживает химические процессы внутри реактора, передавая буквенно-цифровые коды. Некоторые датчики могут передавать данные с ошибками. Если сообщение не содержит ошибок, то четверичная запись суммы восьмеричных цифр в этом сообщении заканчивается на **13**. 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 букв латинского алфавита и цифр, его длина не превышает 30 символов.

#### Формат вывода

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

### Примеры

Входные данные	Выходные данные
4 avb9fvf25f sfew344qd bav994534223 dsf3	2

### Тесты

Входные данные	Выходные данные
4 avb9fvf25f sfew344qd bav994534223 dsf3	2
1 sds342	0
3 fsd	0

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

sdfe fwefs	
4 99999998 1231772 456 65	1
7 a2 a5 a7 a7 sf as a7	3

Решение

```

program szv1;
const
  digits = '1234567';

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 16) = 7 then
      c:=c+1;
    end;
    writeln(c);
end.

```

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

**Задача 6 - Ситуационная задача**

Школьники собирают дрона для исследования реки. Чтобы дрон нормально погружался в воду, нужно повесить на него балласт. В качестве балласта используются пакеты с песком. В наличии у школьников несколько разных пакетов. Известно, какая масса песка помещается в каждый пакет, если песок насыпать нормально. Песок можно утрамбовать, тогда в пакете его будет примерно на 10% больше – в этом случае масса балласта округляется до ближайшего целого в меньшую сторону. Трамбовать песок в пакетах нежелательно. Зная, сколько песка помещается в каждый пакет, какое минимальное количество пакетов потребуется, чтобы набрать балласт, и в скольких из них будет утрамбован песок. Если существуют две комбинации с одинаковым количеством задействованных пакетов, следует вывести ту, где больше пакетов заполнены нормально.

*Формат ввода*

В первой строке программы вводится натуральное число  $N$  ( $N \leq 20$ ) – количество пакетов. Во второй строке указывается целое число  $M$  ( $0 \leq M \leq 20000$  г) – требуемая масса балласта. Далее в  $N$  строчках вводится по одному натуральному числу  $m_i$  ( $m_i \leq 5000$  г) – масса песка, которая помещается в пакет под номером  $i$  при нормальных условиях.

*Формат вывода*

Вывести два целых числа – минимально необходимое количество пакетов, необходимых для набора балласта и количество тех, в которых песок насыпан нормально. Масса балласта должна строго совпадать с требуемой. Если обеспечить такую массу невозможно, вывести 0 0.

**Пример**

Входные данные	Выходные данные
6 61 20 4 3 2 19 17	4 3

**Тесты**

Входные данные	Выходные данные
6 61 20 4 3 2 19 17	4 3
3 31 10 15 5	3 2
5 55 10 10 10	5 0

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

10 10	
6 90 5 10 14 25 1 3	0 0
3 60 20 18 20	3 2

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;

```

Олимпиада школьников «Шаг в будущее»  
Заключительный этап

```
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.
```