

# Технологии дополненной реальности

2022/23 учебный год

## Первый отборочный этап

### Предметный тур. Информатика

#### Первая попытка. Задачи 8–11 класса

##### *Задача П.1.1.1. Три квадрата (15 баллов)*

*Темы: математика, задачи для начинающих.*

##### *Условие*

Фермер владеет участком земли в форме прямоугольника с длинами сторон  $a$  и  $b$ . Недавно фермер понял, что может разбить свой участок на три части так, что каждая часть будет иметь форму квадрата, и решил воспользоваться этой возможностью. Напишите программу, которая найдет площадь каждой части после разбиения.

##### *Формат входных данных*

На вход подается два натуральных числа  $a$  и  $b$  — длины сторон прямоугольника. Числа не превосходят 1000. Каждое число подается в отдельной строке. Гарантируется, что длины сторон таковы, что прямоугольник может быть разбит на три квадрата.

##### *Формат выходных данных*

Требуется вывести через пробел три натуральных числа — площади каждого из участков после разбиения. Числа могут выводиться в произвольном порядке.

##### *Методика проверки*

Программа проверяется на 12-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

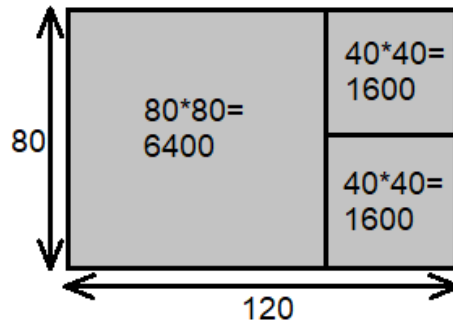
##### *Примеры*

##### *Пример №1*

<b>Стандартный ввод</b>
120 80
<b>Стандартный вывод</b>
6400 1600 1600

### Пояснения к примеру

При заданных размерах прямоугольник может быть разбит на три квадрата так, как показано на рисунке ниже. Обратите внимание, что могут существовать и другие варианты разбиения.



### Пример программы-решения

Ниже представлено решение на языке Python 3.

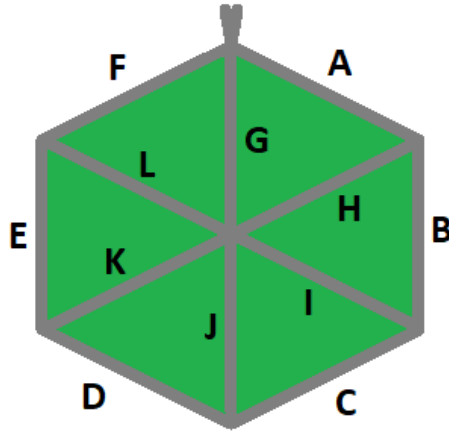
```
1 a, b = map(int, input())
2 if a > b:
3     a, b = b, a
4 s1 = a * a
5 s2 = (a * b - s1) // 2
6 print(s1, s2, s2)
```

### Задача II.1.1.2. Пробежка в шестиугольном парке (18 баллов)

Темы: конструктивное построение, задачи для начинающих.

#### Условие

Иван Иванович совершает пробежки по парку, который имеет форму шестиугольника. В парке 12 аллей, обозначенных символами латинского алфавита от «А» до «L». Схему парка смотрите на рисунке. Длина каждой аллеи ровно 100 м. В парке есть только один вход у перекрестка аллей «А», «F», «G». Иван Иванович хочет начать и закончить пробежку у входа в парк и пробежать ровно  $k$  м. На каждом перекрестке Иван Иванович может повернуть в любую сторону, но он не хочет поворачивать назад.



Напишите программу, которая составит любой маршрут движения, удовлетворяющий указанным требованиям.

### *Формат входных данных*

На вход в подается одно натуральное число  $k$  — желаемая длина маршрута,  $300 \leq k \leq 10000$ . Число  $k$  делится на 100 без остатка.

### *Формат выходных данных*

Требуется вывести строку из  $k/100$  символов, содержащую обозначения аллей в построенном маршруте.

### *Методика проверки*

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
800
<b>Стандартный вывод</b>
FLKEFAHG

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```

1 a=int(input())//100
2 ans=''
3 while a>=6:
4     ans+='AHG'

```

```
5     a-=3
6     sol=['', '', '', 'AHG', 'ABIG', 'ABCJG']
7     print(ans+sol[a])
```

### **Задача П.1.1.3. Знакопеременная сумма (25 баллов)**

Темы: префиксные суммы.

#### **Условие**

Знакопеременной суммой последовательности чисел  $a_1, a_2, \dots, a_k$  называется результат вычисления выражения  $a_1 - a_2 + a_3 - a_4 + a_5 - \dots + (-1)^{k+1}a_k$ . Другими словами, мы складываем все числа в последовательности, но элементы с нечетными номерами мы берем со знаком плюс, а четные со знаком минус.

Задан массив  $x_1, x_2, \dots, x_n$  из  $n$  целых чисел и  $m$  запросов. Каждый запрос содержит по два натуральных числа  $b$  и  $e$ . В ответ на каждый запрос ваша программа должна взять подмассив с номерами элементов от  $b$  до  $e$  включительно и посчитать его знакопеременную сумму  $x_b - x_{b+1} + x_{b+2} - x_{b+3} + \dots + (-1)^{e-b}x_e$

#### **Формат входных данных**

На вход в первой строке подается одно натуральное число  $n$  размер массива чисел,  $1 \leq n \leq 10^5$ . Во второй строке через пробел записаны элемент массива целые числа  $x_1, x_2, \dots, x_n$ . Каждое из чисел не превосходит  $10^6$  по абсолютной величине. Далее в третьей строке записано одно натуральное число  $m$  количество запросов,  $1 \leq m \leq 10^5$ . В каждой из  $m$  последующих строк записано по два числа  $b_i$  и  $e_i$  таких, что  $1 \leq b_i \leq e_i \leq n$ . Каждая пара чисел задает границы подмассива для выполнения одного запроса.

#### **Формат выходных данных**

Требуется вывести через пробел  $m$  целых чисел  $s_1, \dots, s_m$ . Каждое из чисел должно быть равно знакопеременной сумме соответствующего подмассива.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end='')`.

#### **Методика проверки**

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых 10-ти тестах размер массива и количество запросов не превосходят 1000.

## Примеры

### Пример №1

Стандартный ввод
5
7 -4 1 3 2
4
2 2
1 3
2 4
4 5

Стандартный вывод
-4 12 -2 1

### Пояснения к примеру

Для каждого из запросов ответ получается следующим образом:

$$-4 = -4$$

$$7 - (-4) + 1 = 12$$

$$-4 - 1 + 3 = -2$$

$$3 - 2 = 1$$

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 X=list(map(int,input().split()))
3 S=[0]
4 m=1
5 for x in X:
6     S.append(S[-1]+m*x)
7     m=-m
8 m=int(input())
9 ans=''
10 for i in range(3,m+3):
11     b,e=map(int,input().split())
12     s=S[e]-S[b-1]
13     if b%2==0:
14         s=-s
15     print(s,end=' ')
```

### Задача П.1.1.4. Проверка корректности маршрута (25 баллов)

Темы: реализация.

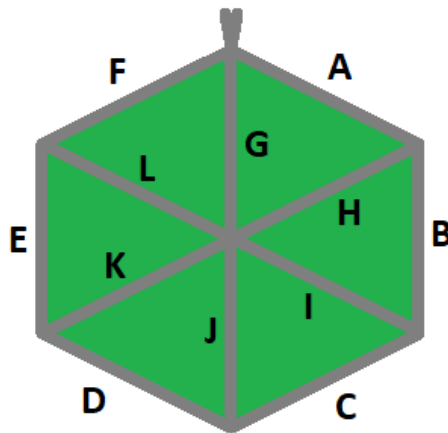
## Условие

В некоторых случаях проверить корректность работы программы гораздо сложнее, чем написать ее. Сейчас у вас будет шанс в этом убедиться. От вас требуется написать программу для проверки корректности ответа второй задачи этой попытки. Напомним ее условие.

Имеется парк в виде шестиугольника с 12-ю аллеями, которые обозначены заглавными символами латиницы. В парке есть только один вход у перекрестка аллей «А», «F», «G». Схема парка приведена ниже. Требуется проверить корректность составленного маршрута движения по этому парку. Маршрут представляется как последовательность символов, представляющих аллеи в том порядке, в котором они были пройдены. Маршрут движения считается корректным если выполняются следующие требования.

- Описание маршрута содержит только символы от «А» до «L».
- Маршрут начинается и заканчивается у входа в парк.
- Запрещено разворачиваться на  $180^\circ$ . В частности, это означает, что начав движение с одного конца аллеи, вы обязательно дойдете до другого ее конца, причем на перекрестке вы должны будете перейти на другую аллею.

На вход вашей программе будет подано несколько описаний маршрутов. Ваша программа должна будет определить, какие из них удовлетворяют указанным требованиям.



## Формат входных данных

На вход в первой строке подается одно натуральное число  $n$  — количество проверяемых маршрутов,  $1 \leq n \leq 20$ . Далее в  $n$  строках записаны сами маршруты. Описание каждого маршрута состоит из последовательности заглавных символов латиницы. Каждое описание не пустое, и содержит не более 100 символов.

## Формат выходных данных

Программа должна вывести строку из  $n$  нулей и единиц. Единица на  $i$ -той позиции означает, что маршрут с номером  $i$  является корректным. В противном случае в этой позиции должен быть записан ноль.

## Методика проверки

Программа проверяется на 5-ти тестах. Прохождение каждого теста оценивается в 5 баллов. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
6 ABCDKHA FMG ABBA ABCEF BCDEF ABCDK
<b>Стандартный вывод</b>
100000

### Пояснения к примеру

Первый маршрут является корректным.

Второй маршрут содержит недопустимое обозначение аллеи.

В третьем маршруте происходит разворот на 180°.

Четвертый маршрут не является связным. После третьего шага он приходит к перекрестку «С», «D», «J» и с него нельзя попасть на аллею «E».

Пятый маршрут начинается не у входа.

Шестой маршрут заканчивается не у входа.

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 def checkway(way):
2     c=0
3     p='Z'
4     for x in way:
5         if x<'A' or x>'L' or x==p:
6             return False
7         p=x
8         num=ord(x)-ord('A')
9         if num>5:
10            if c==6:
11                c=num-6
12            elif c==num-6:
13                c=6
14            else:
15                return False
16     else:
```

```

17         if c==num:
18             c=(c+1)%6
19         elif c==(num+1)%6:
20             c=(c+5)%6
21         else:
22             return False
23     return c==0
24
25     m=int(input())
26     for i in range(m):
27         print(int(checkway(input()))))

```

## Вторая попытка. Задачи 8–11 класса

### Задача П.1.2.1. Стрелки часов (12 баллов)

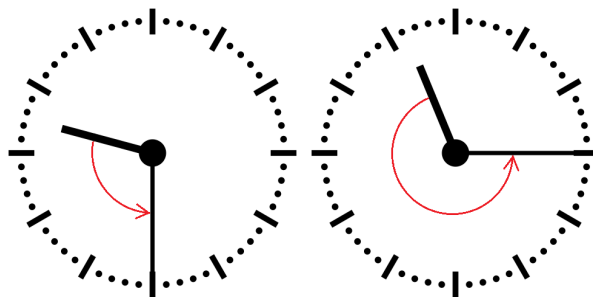
Темы: математика, задачи для начинающих.

#### Условие

Иван Иванович посмотрел на часы и заметил, что минутная и часовая стрелки образуют угол в  $\alpha^\circ$ . С этого момента прошло  $h$  ч и  $m$  мин. Напишите программу, которая вычислит угол между стрелками после истечения этого времени.

Угол измеряется от часовой до минутной стрелки в направлении против хода часов. Если стрелки совпадают, то угол равен нулю. Рассмотрим пример на рисунке. Пусть  $\alpha = 105^\circ$ . В частности, такой угол появляется в 9:30, так как минутная стрелка в этот момент указывает на  $270^\circ$ , а часовая на  $165^\circ$ . Через 1 ч 45 мин на часах будет 11:15. В этот момент времени минутная стрелка указывает на  $0^\circ$ , а часовая на  $112,5^\circ$ . Угол от часовой до минутной стрелки будет равен  $360^\circ - 112,5^\circ = 247,5^\circ$ .

Отметим, что угол  $\alpha = 105^\circ$  появляется и в другие моменты времени, однако, это не повлияет на итоговый ответ.



#### Формат входных данных

На вход в первой строке подается одно целое неотрицательное число  $\alpha$  исходный угол между стрелками,  $0 \leq \alpha \leq 359$ . Во второй строке через пробел на вход подается два числа  $h$  и  $m$  время, прошедшее с момента наблюдения в часах и минутах,  $0 \leq h \leq 11$ ;  $0 \leq m \leq 59$ .



---

## Формат выходных данных

Программа должна вывести одно вещественное число ответ к задаче. Ответ должен быть записан без погрешности.

## Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
105
1 45
<b>Стандартный вывод</b>
247.5

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 a = float(input())
2 h,m = map(int,input().split())
3 a = a - 5.5*(m+h*60)
4 while a<0:
5     a += 360
6 print(a)
```

## Задача II.1.2.2. Номера домов (18 баллов)

Темы: реализация.

### Условие

В поселке Березовом на улице Березовой построено  $n$  домов с номерами от 1 до  $n$ , причем дома с нечетными номерами расположены на одной стороне улицы, а с четными — на другой.

Рано утром дворник вышел к одному из крайних домов на этой улице, который имел номер  $k$ , и начал подметать тротуар, двигаясь от одного края улицы к другому краю. Потом он перешел на противоположную сторону улицы и начал подметать тротуар там, двигаясь назад.

Напишите программу, которая выведет номера домов, мимо которых проходил дворник, по известным числам  $n$  и  $k$ . Для лучшего понимания прочитайте пояснения к примерам.

---

### *Формат входных данных*

На вход в одной строке подается два натуральных числа  $n$  и  $k$  — количество домов и номер одного из крайних домов на улице,  $4 \leq n \leq 100$ . Число  $k$  может принимать одно из четырех значений:  $1, 2, n - 1, n$ .

### *Формат выходных данных*

Ваша программа должна вывести через пробел последовательность номеров домов, в том порядке, в котором их проходил дворник.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end=' ')`.

### *Методика проверки*

Программа проверяется на 36-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
7 6
<b>Стандартный вывод</b>
6 4 2 1 3 5 7

#### *Пример №2*

<b>Стандартный ввод</b>
8 1
<b>Стандартный вывод</b>
1 3 5 7 8 6 4 2

### *Пояснения к примеру*

Рассмотрим первый пример. На улице 7 домов, дворник вышел к дому номер 6. Это означает, что он находится на четной стороне в конце улицы, и далее он будет двигаться к ее началу, проходя мимо домов 6, 4, 2. Потом он перейдет на противоположную сторону к дому номер 1 и пойдет к концу улицы, проходя мимо домов 1, 3, 5, 7.

Во втором примере на улице 8 домов, дворник вышел к дому номер 1. Это означает, что он находится на нечетной стороне в начале улицы, и далее он будет двигаться к ее концу, проходя мимо домов 1, 3, 5, 7. Потом он перейдет на противоположную сторону к дому номер 8 и пойдет к началу улицы, проходя мимо домов 8, 6, 4, 2.

---

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```
1 n,k=map(int,input().split())
2 if k==1:
3     print(*range(1,n+1,2),*range(n-(n%2),0,-2))
4 elif k==2:
5     print(*range(2,n+1,2),*range(n+(n%2)-1,0,-2))
6 elif k%2==1:
7     print(*range(n+(n%2)-1,0,-2),*range(2,n+1,2))
8 else:
9     print(*range(n-(n%2),0,-2),*range(1,n+1,2))
```

### *Задача II.1.2.3. Упорядочивание монет (25 баллов)*

*Темы: реализация, сортировки, теория графов.*

#### *Условие*

В древнем кладе было найдено  $n$  монет различного веса. Каждая из монет была обозначена строчной буквой латиницы. Все обозначения были различными. Монеты были попарно взвешены на чашечных весах. Протокол взвешиваний состоял из  $n(n-1)/2$  строк, каждая строка содержала ровно три символа. Первый и третий символ содержали обозначения монет, а во втором был записан результат сравнения: знак  $<$  или знак  $>$ . Например, запись  $d > b$  означает, что монета  $d$  тяжелее монеты  $b$ .

Взвешивания очень утомили лаборанта, и он просит вас написать программу, которая упорядочит монеты по возрастанию веса.

#### *Формат входных данных*

На вход в первой строке подается одно натуральное число  $n$  — количество монет,  $4 \leq n \leq 26$ . Далее в  $n(n-1)/2$  строках записан протокол взвешиваний. Гарантируется, что протокол является корректным.

#### *Формат выходных данных*

Ваша программа должна вывести одну строку из  $n$  символов. Строка должна содержать обозначения монет в порядке возрастания их веса.

#### *Методика проверки*

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
4 b<x k>b x<k b<d x>d d<k
<b>Стандартный вывод</b>
bdxk

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 cmp=set()
3 for i in range(n*(n-1)//2):
4     cmp.add(input())
5 lst=list({s[0] for s in cmp} | {s[2] for s in cmp})
6 for i in range(len(lst)-1):
7     for j in range(i+1,len(lst)):
8         if lst[j]+'<'+lst[i] in cmp or lst[i]+'>'+lst[j] in cmp:
9             lst[i],lst[j]=lst[j],lst[i]
10 print(''.join(lst))
```

### Задача II.1.2.4. 2–3 дерево (25 баллов)

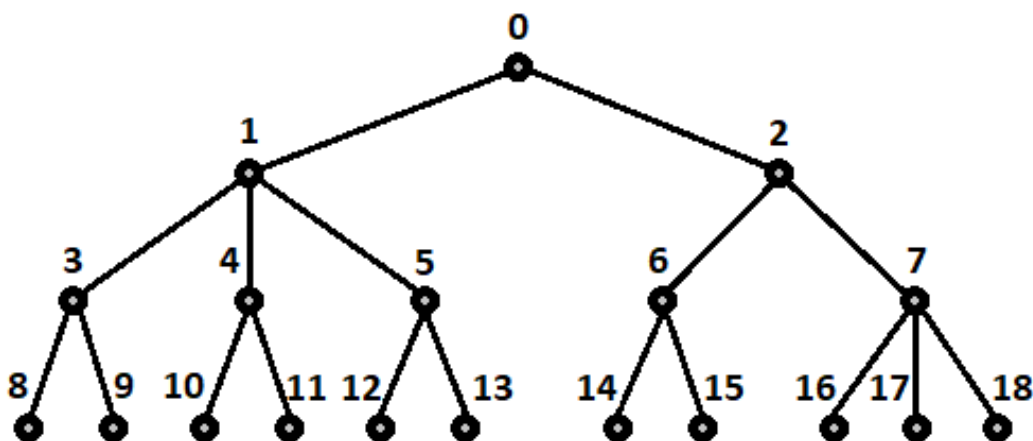
Темы: математика, графы и деревья, реализация.

#### Условие

Упорядоченное корневое дерево называется 2–3 деревом, если для него выполняются следующие условия:

- все узлы, кроме листьев, имеют два или три потомка;
- все листья находятся на одной высоте.

Пример 2–3 дерева приведен на рисунке ниже. Вы должны будете написать программу, которая составит произвольное 2–3 дерево с заданным количеством узлов или определит, что таких деревьев не существует.



### Формат входных данных

На вход в первой строке подается одно натуральное число  $n$  — количество узлов в дереве,  $2 \leq n \leq 100000$ .

### Формат выходных данных

Ваша программа должна вывести описание полученного дерева. Узлы дерева должны иметь номера от 0 до  $n - 1$  и быть упорядоченными сверху вниз и слева направо, как на рисунке. Для каждого узла, кроме корня, требуется указать номер его непосредственного предка. Вывод состоит из последовательности чисел  $a_1, a_2, \dots, a_{n-1}$ , записанных через пробел, где  $a_i$  это непосредственный предок узла с номером  $i$ .

Можно дать и другую интерпретацию. Вывод состоит из последовательности номеров нелистовых узлов, упорядоченных по возрастанию, причем номер каждого узла повторяется столько раз, сколько у него непосредственных потомков.

Если 2–3 дерево с указанным количеством узлов построить невозможно, то требуется вывести  $-1$ .

### Методика проверки

Программа проверяется на 50-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

### Примеры

#### Пример №1

<b>Стандартный ввод</b>
19
<b>Стандартный вывод</b>
0 0 1 1 1 2 2 3 3 4 4 5 5 6 6 7 7 7

## Пример №2

Стандартный ввод
5
Стандартный вывод
-1

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 k=1
3 curc=1
4 maxc=1
5 cnt=[1]
6 while maxc<n:
7     maxc+=3**k
8     curc+=2**k
9     cnt.append(2**k)
10    k+=1
11 if curc>n:
12     print('-1')
13 else:
14     for i in range(1,k):
15         t=min(3*cnt[i-1]-cnt[i],(n-curc)//(2**(k-i)-1))
16         curc+=t*(2**(k-i)-1)
17         for j in range(i,k):
18             cnt[j]+=t*(2**(j-i))
19     m=0
20     for i in range(k-1):
21         p=cnt[i]
22         s=cnt[i+1]
23         while p>0:
24             if s>2*p:
25                 print(m,m,m,end=' ')
26                 s-=3
27             else:
28                 print(m,m,end=' ')
29                 s-=2
30         p-=1
31         m+=1
```

## Третья попытка. Задачи 8–11 класса

### Задача П.1.3.1. Количество нечетных чисел (12 баллов)

Темы: математика, задачи для начинающих.

#### Условие

Задан интервал целых чисел  $[b; e]$ . Вы должны написать программу, чтобы определить, сколько нечетных чисел принадлежит этому интервалу.

---

Обратите внимание, что интервал может быть достаточно большим, и решения, перебирающие все натуральные числа, не будут проходить часть тестов.

### *Формат входных данных*

На вход в одной строке подается два целых числа  $b$  и  $e$  — границы интервала,  $-10^{18} \leq b \leq e \leq 10^{18}$ .

### *Формат выходных данных*

Программа должна вывести одно число — количество нечетных чисел в заданном интервале.

### *Методика проверки*

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
-3 8
<b>Стандартный вывод</b>
6

#### *Пример №2*

<b>Стандартный ввод</b>
0 0
<b>Стандартный вывод</b>
0

### *Пояснения к примеру*

В первом примере указанному интервалу принадлежат шесть нечетных чисел:  $-3, -1, 1, 3, 5, 7$ .

Во втором примере интервал не содержит нечетных чисел.

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```
1 b, e = map(int, input().split())
2 print(((e+e%2)-(b-b%2))/2)
```

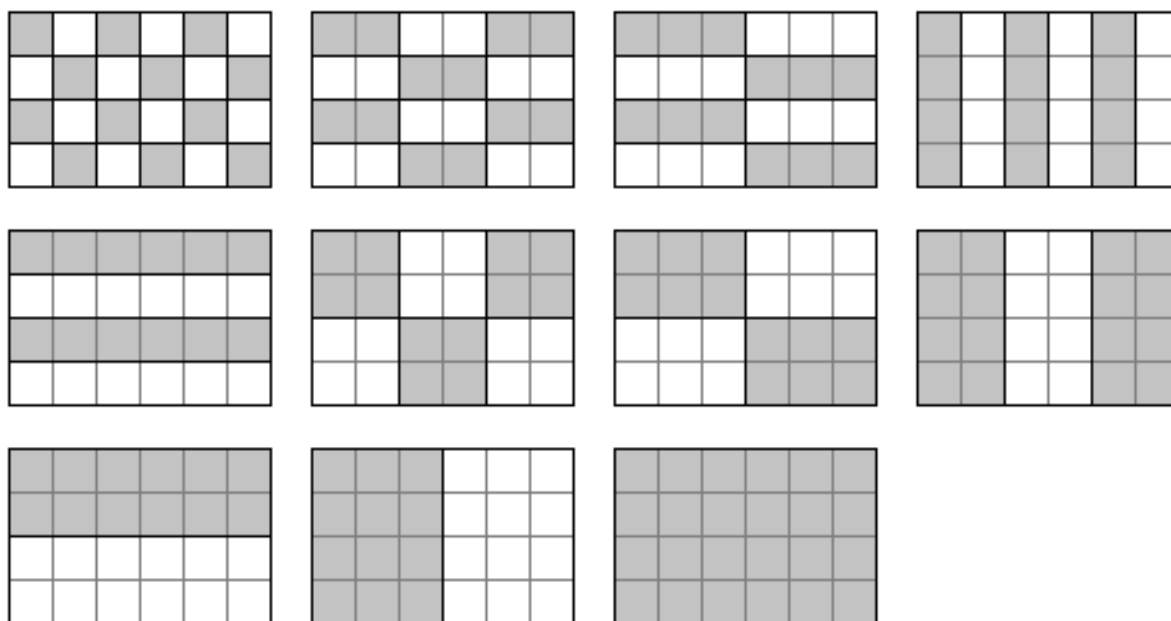
---

## Задача П.1.3.2. Наибольший общий делитель прямоугольников (18 баллов)

Темы: математика, задачи для начинающих.

### Условие

Будем говорить, что прямоугольник  $P$  является делителем прямоугольника  $Q$ , если прямоугольник  $Q$  можно замостить прямоугольниками  $P$ , причем все они должны иметь одинаковую ориентацию. Например, делителями прямоугольника  $6 \times 4$  будут следующие прямоугольники:  $1 \times 1$ ,  $2 \times 1$ ,  $3 \times 1$ ,  $4 \times 1$ ,  $6 \times 1$ ,  $2 \times 2$ ,  $3 \times 2$ ,  $4 \times 2$ ,  $6 \times 2$ ,  $4 \times 3$ ,  $6 \times 4$ . Примеры замощений можно увидеть на рисунке ниже. Обратите внимание, что прямоугольники  $a \times b$  и  $b \times a$  считаются одинаковыми.



Ваша задача заключается в написании программы, которая найдет наибольший общий делитель двух заданных прямоугольников. Из двух прямоугольников большим считается тот, площадь которого больше. Наибольших общих делителей может быть два. В этом случае допускается вывести любой из этих двух прямоугольников.

### Формат входных данных

На вход в двух строках подаются размеры двух прямоугольников. Каждая строка содержит два натуральных числа — длину и ширину прямоугольника. Каждое из чисел не превосходит  $10^{18}$ . Гарантируется, что введенные значения будут таковы, что площадь прямоугольника, который должен получиться в качестве ответа, не превысит  $10^{18}$ .

### Формат выходных данных

Программа должна вывести через пробел два числа — размеры искомого прямоугольника. Числа можно выводить в любом порядке.



---

## Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

## Примеры

### Пример №1

Стандартный ввод
6 4 4 3
Стандартный вывод
4 3

### Пример №2

Стандартный ввод
9 10 15 3
Стандартный вывод
3 5

### Пример №3

Стандартный ввод
3 7 7 3
Стандартный вывод
3 7

### Пример №4

Стандартный ввод
3 7 4 8
Стандартный вывод
1 1

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 def gcd(a,b):
2     while a>0 and b>0:
3         if a>b:
4             a%=b
5         else:
```

```
6         b%=a
7     return a+b
8
9     a1,b1=map(int,input().split())
10    a2,b2=map(int,input().split())
11    a3=gcd(a1,a2)
12    b3=gcd(b1,b2)
13    a4=gcd(a1,b2)
14    b4=gcd(b1,a2)
15    if a3*b3>a4*b4:
16        print(a3,b3)
17    else:
18        print(a4,b4)
```

### **Задача П.1.3.3. Справедливый дележ (25 баллов)**

Темы: реализация.

#### **Условие**

Два купца, живущие в разных городах, в далеком плавание купили несколько видов пряностей, и теперь хотят поделить их. Каждый из купцов будет продавать пряности только в своем городе, и цена каждой пряности в этих городах может отличаться. Купцы сочли, что будет справедливым, если они поделят пряности на две доли так, чтобы суммарная стоимость пряностей первой доли в первом городе была равна суммарной стоимости пряностей второй доли во втором городе. Существует несколько способов дележа, удовлетворяющих этому условию, но купцы хотят выбрать из них такой, при котором они получают максимум денег. Пряности являются сыпучим товаром, поэтому они могут быть поделены в любой пропорции

Рассмотрим пример. Есть три вида пряностей: перец, ваниль и корица. Стоимость всей партии перца в первом и втором городах составляет 120 и 200 условных единиц соответственно. Аналогичная стоимость партии ванили равна 180 и 140 условных единиц, а корицы — 100 и 60 условных единиц. Допустимым способом дележа будет, например, следующий: первый купец возьмет всю ваниль, второй — весь перец, а корицу они поделят поровну. Тогда стоимость доли первого купца в первом городе будет равна  $180 + 100 \cdot 0,5 = 230$ . Стоимость доли второго купца во втором городе составит  $200 + 60 \cdot 0,5 = 230$ . Стоимости долей равны, поэтому такой вариант дележа допустим. Но более выгодным будет другой вариант. Первый купец возьмет всю корицу и  $3/4$  ванили, а второй купец — весь перец и  $1/4$  ванили. Тогда стоимость доли в первом городе составит  $100 + 180 \cdot 0,75 = 235$  и  $200 + 140 \cdot 0,25 = 235$  во втором городе. Таким образом, второй вариант является более предпочтительным.

Напишите программу, которая найдет максимальную стоимость долей, при условии того, что дележ будет справедливым.

#### **Формат входных данных**

На вход в первой строке подается одно натуральное число  $n$  — количество видов пряностей,  $1 \leq n \leq 100$ . Во второй строке через пробел записаны  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$  — цены всех видов пряностей в первом городе. Аналогично в третьей строке записаны числа  $b_1, b_2, \dots, b_n$  — цены всех видов пряностей во втором городе,  $1 \leq a_i, b_i \leq 10^6$ .

---

## Формат выходных данных

Программа должна вывести одно число — максимальную стоимость долей. Это число может быть вещественным. Ответ будет считаться верным, если он отличается от ответа жюри не более чем на  $0,01$ .

## Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пяти тестах  $n \leq 3$ . В первых 15 тестах  $n \leq 10$ . Тесты из условия задачи при проверке не используются.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
3 120 180 100 200 140 60
<b>Стандартный вывод</b>
235.0

### Пример №2

<b>Стандартный ввод</b>
1 100 200
<b>Стандартный вывод</b>
66.66666666666667

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 A=map(float,input().split())
3 B=map(float,input().split())
4 p=list(zip(A,B))
5 p.sort(key=lambda x:x[1]/x[0])
6 s1,s2=0,0
7 i,j=0,n-1
8 while i<=j:
9     if s1<s2:
10         s1+=p[i][0]
11         i+=1
12     else:
13         s2+=p[j][1]
14         j-=1
15 if s1<s2:
```

```

16     s1+=(s2-s1)*p[j+1][0]/(p[j+1][0]+p[j+1][1])
17 else:
18     s1--=(s1-s2)*p[i-1][0]/(p[i-1][0]+p[i-1][1])
19 print(s1)

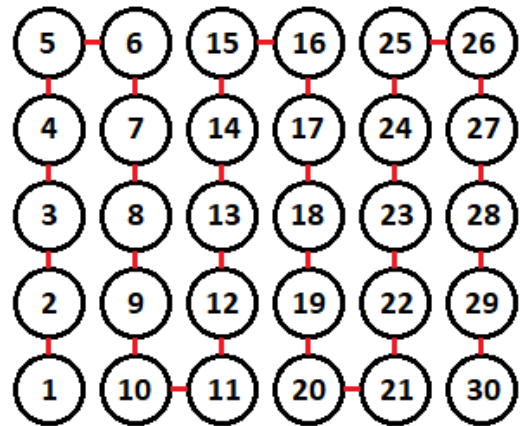
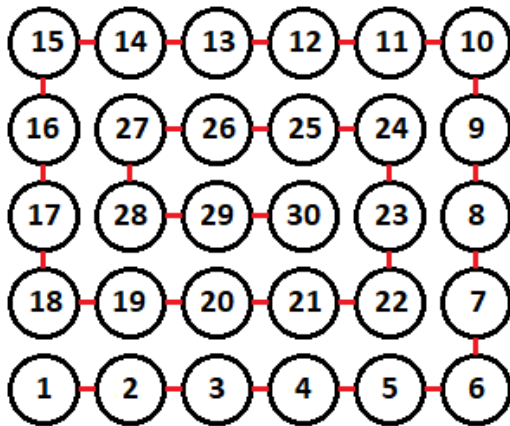
```

### Задача П.1.3.4. Чай с лимоном и сахаром (25 баллов)

Темы: реализация.

#### Условие

На прямоугольном столе в  $n$  рядов расставлены стаканы с чаем, в каждом ряду по  $m$  стаканов. Аня ходит вокруг стола и бросает в каждый стакан по ломтику лимона. Нумерация стаканов на рисунке слева соответствует той последовательности, в которой Аня переходит от одного стакана к другому. Яна ходит вдоль одного края стола туда и обратно, и бросает в каждый стакан кусочек сахара. Нумерация стаканов на рисунке справа соответствует той последовательности, в которой Яна переходит от одного стакана к другому.



Будем считать, что ломтик лимона и кусочек сахара в один стакан девочки бросают ровно за одну секунду. Напишите программу, которая найдет количество стаканов, в которых через  $t$  с лежит и лимон и сахар. В каждом тесте ваша программа должна будет ответить на  $k$  запросов. При этом количество и расположение стаканов на столе единое для всех запросов в одном тесте.

#### Формат входных данных

На вход в первой строке подается два натуральных числа  $n$ ,  $m$  и  $k$  — количество рядов на столе, количество кружек в каждом ряду и количество запросов,  $1 \leq n, m \leq 1000$ ,  $1 \leq k \leq 10^5$ . Во второй строке через пробел записано  $k$  натуральных чисел  $t_1, t_2, \dots, t_k$  — моменты времени, для которых требуется решить задачу,  $1 \leq t_i \leq nm$ . Каждый момент времени может встречаться более 1 раза.

#### Формат выходных данных

Программа должна вывести в одной строке через пробел  $k$  чисел — ответы для каждого из заданных моментов времени.

## Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пятнадцати тестах  $n, m \leq 10$ . Тест из условия задачи при проверке не используется.

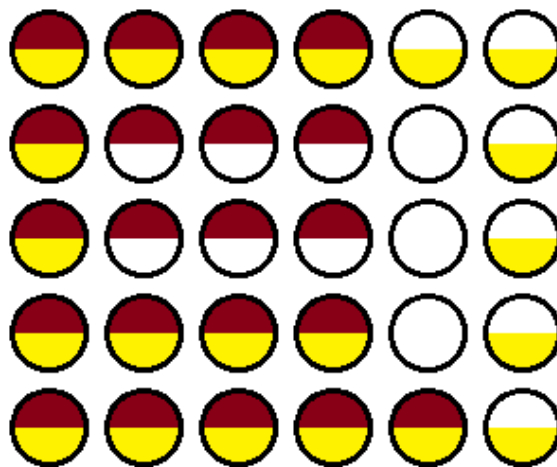
## Примеры

### Пример №1

<b>Стандартный ввод</b>
5 6 4 1 21 30 21
<b>Стандартный вывод</b>
1 15 30 15

### Пояснения к примеру

На рисунке ниже показано решение задачи для теста из условия задачи после двадцать первой секунды. Желтым цветом помечены кружки с лимоном, коричневым — кружки с сахаром. Из рисунка видно, что в 15 чашках есть и лимон, и сахар.



### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n,m,k=map(int,input().split())
2 F=[[0 for i in range(m)] for j in range(n)]
3 A=[0]
4 i1,j1,i2,j2,s=0,0,0,0,0
5 dr=0
6 for i in range(n*m):
7     F[i1][j1]+=1
8     F[i2][j2]+=2
9     if F[i1][j1]==3:
```

```

10     s+=1
11     if (i2!=i1 or j2!=j1) and F[i2][j2]==3:
12         s+=1
13     A.append(s)
14     if dr==0:
15         if j1+1==m or (F[i1][j1+1]&1)==1:
16             dr=1
17     elif dr==1:
18         if i1+1==n or (F[i1+1][j1]&1)==1:
19             dr=2
20     elif dr==2:
21         if j1==0 or (F[i1][j1-1]&1)==1:
22             dr=3
23     else:
24         if i1==0 or (F[i1-1][j1]&1)==1:
25             dr=0
26     if dr==0:
27         j1+=1
28     elif dr==1:
29         i1+=1
30     elif dr==2:
31         j1-=1
32     else:
33         i1-=1
34     if j2%2==0:
35         if i2==n-1:
36             j2+=1
37         else:
38             i2+=1
39     else:
40         if i2==0:
41             j2+=1
42         else:
43             i2-=1
44     for i in input().split():
45         print(A[int(i)])

```

## Четвертая попытка. Задачи 8–11 класса

### Задача П.1.4.1. Сумма элементов списка (12 баллов)

Темы: математика, задачи для начинающих.

#### Условие

Алиса изучает списки в языке Python. По заданию из учебника она написала такую программу.

```

n = int(input())
x = [i%10 for i in range(n)]
print(sum(x))

```

Эта программа читает с консоли натуральное число  $n$  и делает список этой длины, состоящий из чисел от нуля до девяти, которые идут по кругу. Например, для  $n = 25$  список будет иметь вид:

---

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4]

В последней строчке на экран выводится сумма элементов этого списка. Для указанного списка, в частности, будет выведено число 100.

Боб очень недоволен учебником. Он считает, что учебник упускает главное — списки нужны для хранения информации, значимой для работы программы, а это задание можно выполнить, как минимум, без списков, а в идеале — без циклов и условий.

Напишите программу для этого задания, которую Боб сочтет удовлетворительной. Для этого она должна быстро и корректно работать для чисел до  $10^{15}$ .

### *Формат входных данных*

На вход подается одно натуральное число  $n$ , которое не превосходит  $10^{15}$ .

### *Формат выходных данных*

Программа должна вывести одно число — ответ, который напечатала бы приведенная выше программа, если бы она была способна работать со столь большими числами.

### *Методика проверки*

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
25
<b>Стандартный вывод</b>
100

#### *Пример №2*

<b>Стандартный ввод</b>
1000000000000000
<b>Стандартный вывод</b>
4500000000000000

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 return str((n//10)*45+(n%10)*(n%10-1)//2)
```

---

## Задача П.1.4.2. Автокорреляционная функция дискретного сигнала (18 баллов)

Темы: математика, реализация.

### Условие

Автокорреляционная функция часто применяется при анализе сигналов, например, энцефалограммы человека или в радиолокации. Мы будем рассматривать некоторый цифровой сигнал  $a_0, a_1, a_2, \dots, a_{n-1}$ , где каждое значение  $a_i$  равно 1 или  $-1$ . Определим автокорреляционную функцию  $u(t)$  по следующей формуле:

$$u(t) = \sum_{0 \leq i < n-t} a_i a_{i+t}.$$

Другими словами, если сигнал задан в виде списка из  $n$  значений, то чтобы вычислить автокорреляционную функцию в точке  $t$ , требуется взять одну копию списка без первых  $t$  элементов, другую копию списка без последних  $t$  элементов, поэлементно перемножить эти списки, и найти сумму произведений. Рассмотрим пример. Пусть сигнал содержит шесть элементов  $1, 1, -1, 1, -1, 1$ . Найдем  $u(2)$ . Исходная последовательность без первых двух элементов имеет вид  $-1, 1, -1, 1$ . Исходная последовательность без последних двух элементов имеет вид  $1, 1, -1, 1$ . Тогда  $u(2) = (-1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 2$

По такому же принципу можно посчитать и остальные значения для  $t$  от нуля до пяти.

$$u(0) = (1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 6$$

$$u(1) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) + (-1 \cdot 1) + (1 \cdot -1) = -3$$

$$u(3) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) = -1$$

$$u(4) = (-1 \cdot 1) + (1 \cdot 1) = 0$$

$$u(5) = 1 \cdot 1 = 1$$

Напишите программу, которая по заданному дискретному сигналу найдет значения автокорреляционной функции для всех  $t$  от 0 до  $n - 1$ .

### Формат входных данных

На вход в первой строке подается одно натуральное число  $n$  — длина сигнала,  $1 \leq n \leq 100$ . Во второй строке через пробел записаны числа  $a_0, a_1, \dots, a_{n-1}$ , задающие дискретный сигнал. Каждое значение  $a_i$  равно 1 или  $-1$ .

### Формат выходных данных

Программа должна вывести через пробел  $n$  целых чисел — значения автокорреляционной функции  $u(0), u(1), \dots, u(n - 1)$ .

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end=' ')`.



---

## Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
6 1 1 -1 1 -1 1
<b>Стандартный вывод</b>
6 -3 2 -1 0 1

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 a=list(map(int,input().split()))
3 for k in range(n):
4     print(sum([x*y for x,y in zip(a[k:],a[:n-k])]),end=' ')
```

## Задача II.1.4.3. Три фишки (25 баллов)

Темы: игры.

### Условие

Алиса и Боб играют в следующую игру. Имеется игровое поле в виде последовательности клеток, расположенных друг за другом. На поле расположены три фишки, каждая фишка в своей клетке. За один ход каждый игрок должен переместить одну фишку вправо на произвольное ненулевое число клеток. При этом фишка, которой делается ход, не может встать в клетку, где расположена другая фишка или перепрыгнуть через нее. Выигрывает тот игрок, который смог сделать последний ход.

Рассмотрим пример.



Здесь возможны следующие ходы: сместить правую фишку на одну клетку; сместить среднюю фишку на одну клетку; сместить левую фишку на одну, две, три или четыре клетки.

Алиса всегда делает первый ход, а фишки расставляет Боб. Но Боб не хочет побеждать, он хочет, чтобы Алиса нашла выигрышную стратегию. Поэтому он расставляет фишки так, чтобы Алиса могла гарантированно выиграть.

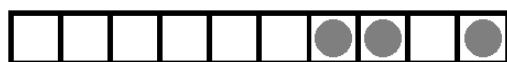
Например, в приведенной выше позиции Алиса должна сместить самую левую фишку на три клетки.



Далее игра зависит от хода Боба. Предположим, он сместит правую фишку на одну клетку. Тогда Алиса в свой ход сместит левую фишку на одну клетку.



Теперь Боб может ходить только средней фишкой. Если он сдвинет ее на одну клетку, то Алиса сдвинет левую фишку на одну клетку.



Бобу остается вновь ходить средней фишкой. Он сдвинет ее на одну клетку, Алиса сдвинет левую фишку на одну клетку и победит.



Для всех других ходов Боба у Алисы также всегда найдется ход, ведущий к победе.

Вы должны написать программу, которая по заданной позиции найдет ход, после которого Алиса сможет победить независимо от дальнейшей игры Боба. Если выигрышных ходов будет несколько, то Алиса может сделать любой из них. Напомним, что исходная позиция будет такой, что найдется как минимум один ход, гарантированно ведущий к победе.

### ***Формат входных данных***

На вход подается строка представляющая игровое поле. Пустая клетка в строке обозначена нулем, клетка с фишкой обозначена единицей. Длина строки не превосходит 1000 символов. В строке ровно три единицы.

### ***Формат выходных данных***

Программа должна вывести строку, представляющую игровое поле после хода Алисы, в том же формате, в котором она поступает на вход.

### ***Методика проверки***

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

Стандартный ввод
0100001010
Стандартный вывод
0000101010

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 x=list(reversed(input()))
2 m=[0,0,0]
3 k=0
4 for c in x:
5     if c=='0':
6         m[k]+=1
7     elif k<2:
8         k+=1
9     else:
10        break
11 if m[0]>m[2]:
12     x[m[0]]='0'
13     x[m[2]]='1'
14 else:
15     x[m[0]+m[1]+m[2]+2]='0'
16     x[2*m[0]+m[1]+2]='1'
17 print(''.join(reversed(x)))
```

### Задача II.1.4.4. Выбор купюр (25 баллов)

Темы: реализация.

#### Условие

В денежной системе Бурляндии выпускаются банкноты всех номиналов от  $a$  до  $2a$  включительно. У Алисы в бумажнике есть ровно одна банкнота каждого номинала. Алиса хочет сделать покупку ценой  $b$  и расплатиться без сдачи. Кроме того, Алиса хочет, чтобы количество потраченных банкнот было как можно меньшим. Напишите программу, которая поможет Алисе выбрать банкноты так, чтобы сумма их номиналов была равна  $b$ , а их количество было наименьшим среди возможных. Если указанным условиям удовлетворяет несколько наборов банкнот, то ваша программа может вывести любой из них.

#### Формат входных данных

На вход в одной строке подается два натуральных числа  $a$  и  $b$  — минимальный из номиналов купюр и требуемая сумма,  $1 \leq a \leq 100000$ . Гарантируется, что для заданной суммы  $b$  существует способ получить ее из имеющихся купюр.

---

## Формат выходных данных

Программа должна вывести в одной строке через пробел номиналы всех банкнот, которые потребуются для оплаты. **Все номиналы должны быть упорядочены по возрастанию.**

## Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
10 99
<b>Стандартный вывод</b>
10 15 17 18 19 20

## Пояснения к примеру

Сумма чисел, указанных в ответе, равна 99, и все числа лежат в диапазоне от 10 до 20 включительно. При этом сумма номиналов пяти самых ценных банкнот меньше чем 99, поэтому оплатить указанную сумму пятью или меньшим числом банкнот невозможно. Однако другие варианты получения требуемой суммы шестью банкнотами возможны, например, 13 14 15 18 19 20. Такой ответ тоже будет засчитан.

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 a,b=map(int,input().split())
2 up,dn,k=0,0,0
3 while up<b:
4     up+=2*a-k
5     dn+=a+k
6     k+=1
7
8 if a>k-1:
9     t=(b-dn)//(a-k+1)
10    r=(b-dn)%(a-k+1)
11 else:
12    t,r=0,0
13 if t==k:
14    print(*range(2*a-t+1,2*a+1))
15 else:
16    print(*range(a,a+k-t-1),a+k-1-t+r,*range(2*a-t+1,2*a+1))
```

# Предметный тур. Математика

## Первая попытка. Задачи 8–9 класса

### Задача П.2.1.1. (25 баллов)

Темы: вероятность, несовместные события.

#### Условие

Участникам «математического марафона» предлагается для решения  $n$  задач, из которых  $k$  имеют повышенную сложность. Задачи решаются по одной в случайном порядке. Участник получает следующую задачу лишь после того, как решил предыдущую. Решив задачу повышенной сложности, участник переходит на следующий уровень. Найти вероятность того, что для выхода на следующий уровень участнику придется решить не более трех задач. Ответ округлить до сотых.

#### Решение

Событие  $A$ , при котором участник решил не более трёх задач до выхода на следующий уровень, можно разбить на три несовместных события:

- $A_1$  — участник решил ровно одну задачу;
- $A_2$  — участник решил ровно две задачи;
- $A_3$  — участник решил ровно три задачи.

В первом случае участник решил одну из  $k$  сложных задач, а всего он мог выбрать одну из  $n$  задач. Поэтому:

$$P(A_1) = \frac{k}{n}.$$

Во втором случае участник сначала решил простую, а затем сложную задачу. Всего способов выбрать пару задач:  $n(n-1)$ . Всего способов выбрать сначала простую, а затем сложную задачу:  $(n-k)k$ . Поэтому:

$$P(A_2) = \frac{(n-k)k}{n(n-1)}.$$

В третьем случае участник сначала решил две простых, а затем сложную задачу. Всего способов выбрать тройку задач:  $n(n-1)(n-2)$ . Всего способов выбрать сначала две простых, а затем сложную задачу:  $(n-k)(n-k-1)k$ . Поэтому:

$$P(A_3) = \frac{(n-k)(n-k-1)k}{n(n-1)(n-2)}.$$

Таким образом, искомая вероятность:

$$P(A) = P(A_1) + P(A_2) + P(A_3) = \frac{k}{n} + \frac{(n-k)k}{n(n-1)} + \frac{(n-k)(n-k-1)k}{n(n-1)(n-2)}.$$

Погрешность 1%.

## Диапазоны

Величина	min	max	Шаг
$n$	10	20	1
$k$	3	7	1

Ответ:  $\frac{k}{n} + \frac{(n-k)k}{n(n-1)} + \frac{(n-k)(n-k-1)k}{n(n-1)(n-2)}$ .

### Задача П.2.1.2. (20 баллов)

Темы: комбинаторика.

#### Условие

В вазе лежат конфеты, среди которых нет одинаковых. После обеда мама разрешила своим двум детям взять по три конфеты. Сначала одну конфету берет младший, потом одну конфету берет старший, затем одну конфету опять берет младший и т. д. Известно, что у младшего ребенка число вариантов выбора на  $n$  больше, чем у старшего. Сколько конфет лежало в вазе до того, как дети стали их брать?

#### Решение

Пусть  $k$  — количество конфет в вазе.

Младший выбирает сначала одну из  $k$  конфет, затем одну из  $k-2$ , и, наконец, одну из  $k-4$ . Всего вариантов у младшего ребёнка  $k(k-2)(k-4)$ .

Старший выбирает сначала одну из  $k-1$  конфет, затем одну из  $k-3$ , и, наконец, одну из  $k-5$ . Всего вариантов у старшего ребёнка  $(k-1)(k-3)(k-5)$ .

По условию:

$$k(k-2)(k-4) - (k-1)(k-3)(k-5) = n.$$

Раскрыв скобки, получаем квадратное уравнение:

$$3k^2 - 15k + 15 - n = 0.$$

Решая его и отбрасывая отрицательный корень, находим:

$$k = \frac{15 + \sqrt{45 + 12n}}{6}.$$

## Диапазоны

$$n = 3(k^2 - 5k + 5),$$

где  $k = 15, \dots, 30$ .

Ответ:  $\frac{15 + \sqrt{45 + 12n}}{6}$ .

### Задача II.2.1.3. (30 баллов)

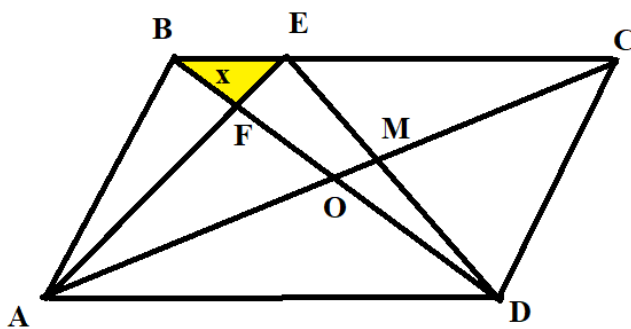
Темы: геометрия, площадь, подобные треугольники.

#### Условие

В параллелограмме  $ABCD$  диагонали пересекаются в точке  $O$ . На стороне  $BC$  отмечена точка  $E$ . Диагональ  $BD$  пересекает отрезок  $AE$  в точке  $F$  так, что площадь треугольника  $ABF$  равна  $a$ . Диагональ  $AC$  пересекает отрезок  $DE$  в точке  $M$  так, что площадь треугольника  $DMC$  равна  $b$ . Найдите площадь треугольника  $BEF$ , если площадь четырёхугольника  $EFOM$  равна  $c$ . Ответ округлите до сотых.

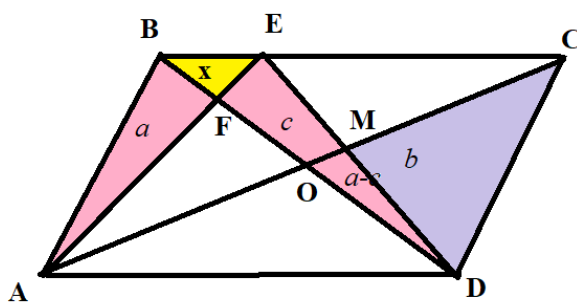
#### Решение

Обозначим искомую площадь за  $x$ .



Рассмотрим треугольники  $ABE$  и  $DBE$ . Их площади совпадают, следовательно,  $a + x = x + c + S_{DOM}$ . Отсюда:

$$S_{DOM} = a - c.$$



Поскольку  $S_{ABO} = S_{DOC} = b + a - c$ , то  $S_{AFO} = S_{ABO} - a = b - c$ .

Теперь найдём площадь треугольника  $AFD$ :

$$S_{AFD} = S_{AFO} + S_{AOD} = S_{AFO} + S_{OCD} = (b - c) + (a - c + b) = 2(b - c) + a.$$

Треугольники  $BEF$  и  $FED$  имеют одинаковую высоту (из вершины  $E$ ). Поэтому их площади относятся как основания:

$$\frac{x}{a} = \frac{BF}{FD}.$$

---

Треугольники  $ABF$  и  $AFD$  также имеют общую высоту (из вершины  $A$ ).

Следовательно, отношение их оснований равно отношению их площадей:

$$\frac{BF}{FD} = \frac{a}{2(b-c) + a}.$$

Сопоставляя два последних равенства, находим:

$$x = \frac{a^2}{2(b-c) + a}.$$

Погрешность 1%.

### *Диапазоны*

Величина	min	max	Шаг
$a$	7	15	1
$b$	8	16	1
$c$	1	6	1

Ответ:  $\frac{a^2}{2(b-c) + a}$ .

### **Задача II.2.1.4. (25 баллов)**

*Темы: алгебра, система линейных уравнений, задача на работу.*

#### **Условие**

Петя и Вася в первый день хакатона совместно за  $n$  часов написали программу, решающую предложенную задачу. Через каждые два часа работы над программой мальчики делали короткий перерыв. При этом производительность после перерыва у Пети становилась на 20%, а у Васи — на 10% меньше производительности до перерыва. Во второй день Петя и вовсе потерял интерес к задачам, а Васю, наоборот, заинтересовала предложенная тематика. Поэтому производительность Пети снизилась в  $k$  раз, а у Васи она увеличилась в  $k$  раз (относительно начала первого дня). Производительность мальчиков падала через каждые два часа так же, как и в первый день. В итоге во второй день они потратили  $n - 2$  часа на совместное написание кода того же объёма, что и в первый день. Во сколько раз в первый день изначальная производительность Васи отличалась от изначальной производительности Пети? Ответ округлите до сотых.

#### **Решение**

Пусть  $x$  — производительность Пети,  $y$  — производительность Васи. По условию задачи составим систему уравнений:

$$\begin{cases} 2(x + y) + 2(0,8x + 0,9y) + (n - 4)(0,8^2x + 0,9^2y) = 1, \\ 2\left(\frac{x}{k} + ky\right) + (n - 4)\left(\frac{0,8x}{k} + 0,9ky\right) = 1. \end{cases}$$



Приводя подобные слагаемые, имеем:

$$\begin{cases} (3,6 + 0,8^2(n-4))x + (3,8 + 0,9^2(n-4))y = 1, \\ \frac{2+0,8(n-4)}{k}x + (2 + 0,9(n-4))ky = 1. \end{cases}$$

Разделив каждое уравнение на  $x$ , находим:

$$\begin{cases} 3,6 + 0,8^2(n-4) + (3,8 + 0,9^2(n-4))\frac{y}{x} = \frac{1}{x}, \\ \frac{2+0,8(n-4)}{k} + (2 + 0,9(n-4))k\frac{y}{x} = \frac{1}{x}. \end{cases}$$

Вычитая уравнения, получаем:

$$(3,8 + 0,9^2(n-4) - 2k - 0,9k(n-4))\frac{y}{x} = \frac{2 + 0,8(n-4)}{k} - 3,6 - 0,8^2(n-4).$$

Отсюда:

$$\frac{y}{x} = \frac{\frac{2+0,8(n-4)}{k} - 3,6 - 0,8^2(n-4)}{3,8 + 0,9^2(n-4) - 2k - 0,9k(n-4)}$$

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$n$	4,5	5	0,5
$k$	2	5	1

Ответ:  $\frac{\frac{2+0,8(n-4)}{k} - 3,6 - 0,8^2(n-4)}{3,8 + 0,9^2(n-4) - 2k - 0,9k(n-4)}$ .

## Первая попытка. Задачи 10–11 класса

### Задача II.2.2.1. (20 баллов)

Темы: теория вероятностей, формула Байеса.

#### Условие

Дистрибьютор покупает компьютеры на трех фабриках. Первая фабрика подготовила к отгрузке  $m_1$  компьютеров,  $k_1$  из которых имеют брак. Вторая фабрика —  $m_2$  компьютеров, среди которых  $k_2$  бракованных. Третья фабрика —  $m_3$  компьютеров, среди которых  $k_3$  бракованных. Дистрибьютор приобрел у каждой фабрики по одному компьютеру. Оказалось, что компьютер, купленный покупателем у дистрибьютора, бракованный. Какова вероятность того, что этот компьютер изготовлен  $j$ -й фабрикой? Ответ округлить до сотых.

---

### Решение

Пусть событие  $A$  — купленный компьютер бракованный, событие  $H_i$  — купленный компьютер изготовлен на  $i$ -й фабрике. В задаче спрашивается, чему равна условная вероятность  $P(H_j|A)$ .

Имеем для всех  $i = 1, 2, 3$ :

$$P(H_i) = \frac{1}{3}, \quad P(A|H_i) = \frac{k_i}{m_i}.$$

Тогда по формуле Байеса

$$P(H_j|A) = \frac{P(H_j)P(A|H_j)}{\sum_{i=1}^3 P(H_i)P(A|H_i)} = \left(\frac{k_j}{m_j}\right) / \left(\frac{k_1}{m_1} + \frac{k_2}{m_2} + \frac{k_3}{m_3}\right).$$

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$m_1$	6	8	1
$k_1$	2	3	1
$m_2$	8	10	1
$k_2$	2	3	1
$m_3$	7	9	1
$k_3$	1	2	1
$j$	1	3	1

Ответ:  $\left(\frac{k_j}{m_j}\right) / \left(\frac{k_1}{m_1} + \frac{k_2}{m_2} + \frac{k_3}{m_3}\right)$ .

### Задача П.2.2.2. (25 баллов)

Темы: комбинаторика, биномиальные коэффициенты.

### Условие

На переэкзаменовку пришло  $a$  студентов 1-го курса и  $b$  студентов 2-го курса. Преподаватель пригласил в аудиторию половину студентов 2-го курса и несколько студентов 1-го курса. Пока студенты готовились, преподаватель подсчитал, что существует  $c$  способов вызвать студентов таким образом (то есть половину пришедших второкурсников и количество вызванных им первокурсников). Какое наибольшее возможное число студентов 1-го курса преподаватель мог вызвать, чтобы полученное им число вариантов не изменилось?

### Решение

Пусть  $k$  — число вызванных преподавателем первокурсников. Тогда число способов вызвать половину второкурсников и  $k$  первокурсников:

$$c = C_b^{b/2} C_a^k.$$

Зная число сочетаний  $C_a^k = \frac{c}{C_b^{b/2}}$  и число  $a$ , найдём  $k \leq a/2$  (можно, например, воспользоваться треугольником Паскаля).

Так как  $C_a^k = C_a^{a-k}$  и  $C_a^k \neq C_a^m$  при  $m \neq k$  и  $m \neq a - k$ , то полученное преподавателем число вариантов не изменится, если вызывать  $a - k$  первокурсников.

### Диапазоны

Величина	min	max	Шаг
$a$	9	12	1
$b$	6	12	2
$k$	3	$\lceil a/2 \rceil - 1$	1
$c$	$C_b^{b/2} C_a^k$		

Ответ:  $a - k$ .

### Задача II.2.2.3. (30 баллов)

Темы: стереометрия, исследование на экстремум.

### Условие

Шар радиуса  $r$  см составлен из двух полушаров: из меди плотности  $8,9$  г/см<sup>3</sup> и олова плотности  $7,3$  г/см<sup>3</sup>. Из этого шара выпилили деталь в форме прямоугольного параллелепипеда наибольшего объёма, стороны основания которого относятся как  $k:1$ . Найти массу опилок (то есть всех частей, оставшихся от шара после выпиливания детали).

Принять  $\pi = 3,1416$  и выразить ответ в г (с точностью до 1 г).

### Решение

Выпиленная деталь представляет собой прямоугольный параллелепипед, вписанный в шар. Её объём:

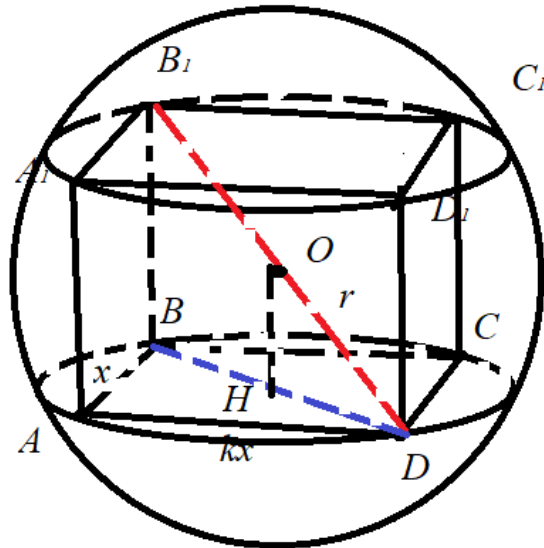
$$V = kx^2 \cdot 2OH.$$

Из треугольников  $ODH$  и  $ABD$  находим:

$$OH^2 = OD^2 - HD^2 = r^2 - \left(\frac{BD}{2}\right)^2 = r^2 - \frac{x^2 + (kx)^2}{4} = \frac{4r^2 - x^2(k^2 + 1)}{4}.$$

Тогда:

$$V = kx^2 \cdot 2\sqrt{\frac{4r^2 - x^2(k^2 + 1)}{4}} = kx^2 \sqrt{4r^2 - x^2(k^2 + 1)}.$$



Найдём максимум функции  $V(x)$ . Её производная:

$$V'(x) = \frac{2kx(4r^2 - x^2(k^2 + 1)) - kx^3(k^2 + 1)}{\sqrt{4r^2 - x^2(k^2 + 1)}} = kx \frac{8r^2 - 3(k^2 + 1)x^2}{\sqrt{4r^2 - (k^2 + 1)x^2}}$$

на промежутке  $(0, \frac{2r}{\sqrt{k^2+1}})$  имеет единственный корень:

$$x = \frac{2r\sqrt{2}}{\sqrt{3(k^2 + 1)}}.$$

Исследование знаков производной показывает, что это точка максимума, а наибольший объём равен:

$$V_{max} = \frac{16kr^3}{3\sqrt{3}(k^2 + 1)}.$$

Объём опилок равен:

$$V_o = \frac{4\pi r^3}{3} - \frac{16kr^3}{3\sqrt{3}(k^2 + 1)} = \frac{4r^3}{3} \left( \pi - \frac{4k}{\sqrt{3}(k^2 + 1)} \right).$$

Поскольку половина этого объёма сделана из олова (плотности  $\rho_1 = 7,3 \text{ г/см}^3$ ), а половина — из меди (плотности  $\rho_2 = 8,9 \text{ г/см}^3$ ), то масса опилок равна:

$$M_o = \frac{V_o}{2}\rho_1 + \frac{V_o}{2}\rho_2 = V_o \frac{\rho_1 + \rho_2}{2} = 8,1V_o = 10,8r^3 \left( \pi - \frac{4k}{\sqrt{3}(k^2 + 1)} \right).$$

Погрешность 1.

### Диапазоны

Величина	min	max	Шаг
$r$	2	6	0,2
$k$	2	10	1

**Ответ:**  $10,8r^3 \left( \pi - \frac{4k}{\sqrt{3}(k^2 + 1)} \right).$

### Задача П.2.2.4. (25 баллов)

Темы: алгебра, функциональное уравнение, линейная система.

#### Условие

Найти значение  $f(1)$  для функции  $f$ , удовлетворяющей при всех  $x \neq a$  уравнению

$$f(x) + kx \cdot f\left(\frac{ax+b}{x-a}\right) = n.$$

Записать ответ с точностью до 0,01.

#### Решение

Пусть  $y = \frac{ax+b}{x-a}$ . Заметим, что  $y \neq a$ . Поэтому выполнено равенство:

$$f(y) + kyf\left(\frac{ay+b}{y-a}\right) = n.$$

Так как:

$$x = \frac{ay+b}{y-a},$$

получаем:

$$f\left(\frac{ax+b}{x-a}\right) + k\frac{ax+b}{x-a}f(x) = n.$$

Положим  $X = f(x)$ ,  $Y = f\left(\frac{ax+b}{x-a}\right)$ . Тогда имеем систему уравнений:

$$\begin{cases} X + kxY = n, \\ Y + kyX = n. \end{cases}$$

Решая эту линейную относительно переменных  $X$  и  $Y$  систему, получаем:

$$X = \frac{n(1-kx)}{1-k^2xy}.$$

Учитывая, что при  $x = 1$  будет  $y = \frac{a+b}{1-a}$ , находим  $f(1) = \frac{n(k-1)(1-a)}{k^2(a+b) - (1-a)}$ .

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$a$	20	50	10
$b$	2	5	1
$k$	2	5	1
$n$	100	200	20

Ответ:  $\frac{n(k-1)(1-a)}{k^2(a+b) - (1-a)}$ .

---

## Вторая попытка. Задачи 8–9 класса

### Задача П.2.3.1. (20 баллов)

Темы: теория вероятности, дополнительное событие.

#### Условие

Участникам олимпиады предлагается для решения  $n$  задач. Вася может решить каждую из задач с вероятностью  $p$ . Найти вероятность того, что Вася решит хотя бы одну задачу. Ответ округлить до сотых.

#### Решение

Пусть  $A$  — событие, при котором Вася решил хотя бы одну задачу. Вычислим вероятность дополнительного события  $\bar{A}$  — Вася не решил ни одну задачу:

$$P(\bar{A}) = (1 - p)^n.$$

Тогда:

$$P(A) = 1 - P(\bar{A}) = 1 - (1 - p)^n.$$

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$n$	4	10	1
$p$	0,1	0,5	0,1

Ответ:  $1 - (1 - p)^n$ .

### Задача П.2.3.2. (25 баллов)

Темы: комбинаторика.

#### Условие

В спортивном семейном лагере отдыхают  $n$  семей, каждая из которых состоит из 5 человек: отца, матери и троих детей. Для предстоящих соревнований с представителями другого спортивного лагеря надо составить команду из пяти человек, в которой должен быть один мужчина, одна женщина, трое детей, но не должно быть родственников. Сколькими способами это можно сделать?

#### Решение

Сначала выбираем мужчину, для этого имеется  $n$  способов. Далее выбираем женщину из оставшихся  $n - 1$  семей. Далее выбираем детей из оставшихся  $n - 2$  семей:

---

$3n - 6$  способов выбрать первого,  $3n - 9$  способов — второго (число способов уменьшается на 3, так как запрещено выбирать братьев или сестёр уже выбранного ребёнка),  $3n - 12$  способов — третьего.

Получается  $N = n(n - 1)(3n - 6)(3n - 9)(3n - 12)$  вариантов. Однако перестановки выбранных детей не дают новой команды, поэтому полученное число нужно разделить на  $3!$ . Итого имеется:

$$\frac{N}{3!} = \frac{4.5 \cdot n!}{(n - 5)!}$$

вариантов.

### *Диапазоны*

Величина	min	max	Шаг
$n$	5	25	1

Ответ:  $\frac{4.5 \cdot n!}{(n - 5)!}$ .

### **Задача II.2.3.3. (25 баллов)**

*Темы: планиметрия, прямоугольный треугольник, вписанная окружность.*

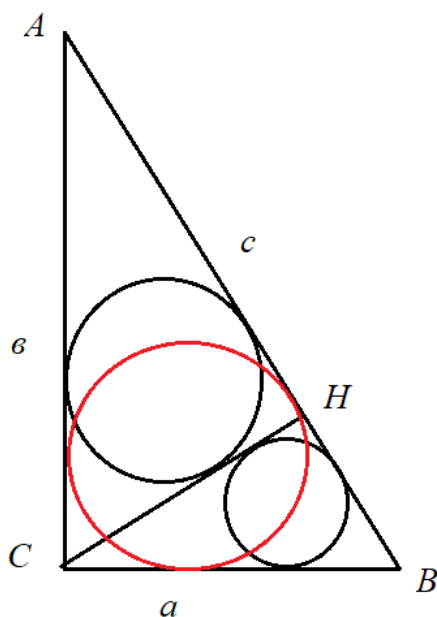
#### **Условие**

В прямоугольном треугольнике ABC из вершины прямого угла проведена высота CH. В треугольник ABC вписана окружность радиуса  $R$ , а в треугольник ACH — окружность радиуса  $r$ . Найти радиус окружности, вписанной в треугольник BCH.

Ввести ответ с точностью до 0,01.

#### **Решение**

Положим  $a = BC$ ,  $b = AC$ ,  $c = AB$ ,  $x$  — искомый радиус.



Из подобия треугольников АСН и АВС имеем  $\frac{r}{R} = \frac{b}{c}$ , следовательно:

$$b = c \frac{r}{R}.$$

Из подобия треугольников ВСН и АВС будет  $\frac{x}{R} = \frac{a}{c}$ , значит:

$$a = c \frac{x}{R}.$$

По теореме Пифагора будет  $a^2 + b^2 = c^2$ , то есть:

$$\left(c \frac{x}{R}\right)^2 + \left(c \frac{r}{R}\right)^2 = c^2.$$

Отсюда находим:

$$x = \sqrt{R^2 - r^2}$$

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$r$	1	4	0,5
$R$	6	12	0,5

Ответ:  $\sqrt{R^2 - r^2}$ .

### Задача II.2.3.4. (30 баллов)

Темы: алгебра, поиск шаблона.



### Условие

Развозчик пиццы каждому адресату отдавал половину имеющихся пицц и ещё половину пиццы (каждый адресат получал целое число пицц). Все пиццы были розданы по  $k$  адресам. Сколько изначально пицц было у развозчика?

### Решение

Пусть у развозчика  $n$  пицц.

Первому адресату он отдаст  $\frac{n}{2} + \frac{1}{2} = \frac{n+1}{2}$  пицц. У него останется  $n - \frac{n+1}{2} = \frac{n-1}{2}$  пицц.

Второму адресату он отдаст  $\frac{n-1}{4} + \frac{1}{2} = \frac{n+1}{4}$  пицц. У него останется  $\frac{n-1}{2} - \frac{n+1}{4} = \frac{n-3}{4}$  пицц.

Третьему адресату он отдаст  $\frac{n-3}{8} + \frac{1}{2} = \frac{n+1}{8}$  пицц. У него останется  $\frac{n-3}{4} - \frac{n+1}{8} = \frac{n-7}{8}$  пицц. И так далее.

Имеем:  $k$ -му адресату он отдаст  $\frac{n+1}{2^k}$  пицц. Так как пиццы у него закончатся, то последний адресат должен получить одну пиццу, то есть:  $\frac{n+1}{2^k} = 1$ .

Следовательно,  $n = 2^k - 1$ .

### Диапазоны

Величина	min	max	Шаг
$k$	10	20	1

Ответ:  $2^k - 1$ .

## Вторая попытка. Задачи 10–11 класса

### Задача П.2.4.1. (25 баллов)

Темы: теория вероятностей, формула полной вероятности.

### Условие

В районной «математической регате» участвует  $n$  команд, причём физико-математический лицей выставил две команды. По правилам соревнований команды случайным образом разбиваются на игровые пары. В каждой паре определяется проигравшая команда, которая выбывает из турнира, и победившая, которая выходит в следующий тур. На очередном туре команды снова распределяются на пары случайным образом. В каждой встрече вероятность выигрыша и поражения для любой команды равна 0,5. Какова вероятность того, что не позднее  $k$ -го тура команды физико-математического лицея сыграют друг с другом? Ответ округлить до тысячных.

## Решение

Пусть  $A$  — событие, при котором команды физико-математического лица (назовём их «команда 1» и «команда 2») сыграют вместе не позднее  $k$ -го тура;  $A_1$  — событие, при котором эти команды встретятся в первом туре. По формуле полной вероятности:

$$P(A) = P(A|A_1)P(A_1) + P(A|\bar{A}_1)P(\bar{A}_1) = P(A_1) + P(A|\bar{A}_1)(1 - P(A_1))$$

(мы воспользовались тем, что  $P(A|A_1) = 1$ ).

Обозначим через  $p_{i,j}$  вероятность события, при котором команды 1 и 2 встретились не позднее  $j$ -го тура, если всего участвует  $2^i$  команд. Тогда формулу, полученную выше, можно записать так:

$$p_{m,k} = p_{m,1} + P(A|\bar{A}_1)(1 - p_{m,1}).$$

Если заранее известно, что произошло событие  $\bar{A}_1$ , то события «команда 1 выиграла» и «команда 2 выиграла» независимы. Тогда вероятность выхода обеих команд во второй тур равна  $0,5^2 = 0,25$ . Следовательно, по формуле полной вероятности имеем:

$$P(A|\bar{A}_1) = p_{m-1,k-1} \cdot 0,25 + 0 \cdot (1 - 0,25) = p_{m-1,k-1} \cdot 0,25.$$

Заметим ещё, что:

$$p_{i,1} = \frac{2^i/2}{C_{2^i}^2} = \frac{2^i}{2^i(2^i - 1)} = \frac{1}{2^i - 1}.$$

Таким образом, получаем рекуррентную формулу:

$$p_{m,k} = \frac{1}{2^m - 1} + 0,25p_{m-1,k-1} \left(1 - \frac{1}{2^m - 1}\right).$$

Применяя последовательно это равенство, находим:

$$\begin{aligned} p_{m,k} &= \frac{1}{2^m - 1} + \frac{2(2^{m-1} - 1)}{2^m - 1} \frac{1}{4} \left( \frac{1}{2^{m-1} - 1} + \frac{2(2^{m-2} - 1)}{2^{m-1} - 1} \frac{1}{4} (\dots) \right) = \\ &= \frac{1}{2^m - 1} + \frac{1}{2} \frac{1}{2^m - 1} + \frac{1}{4} \frac{1}{2^m - 1} + \dots = \frac{1}{2^m - 1} \sum_{i=0}^{k-1} \frac{1}{2^i} = \frac{2}{2^m - 1} \left(1 - \frac{1}{2^k}\right). \end{aligned}$$

Погрешность 0,1%.

## Диапазоны

Величина	min	max	Шаг
$n$	$2^m$		
$m$	5	10	1
$k$	3	4	1

**Ответ:**  $\frac{2}{2^m - 1} \left(1 - \frac{1}{2^k}\right)$ .

---

### Задача II.2.4.2. (20 баллов)

Темы: комбинаторика.

#### Условие

В доме культуры работают 3 различные спортивные секции. Известно, что во второй секции занимается на 3 человека больше, чем в первой, и на 3 человека меньше, чем в третьей, причем спортсмены в каждой секции имеют примерно одинаковый уровень подготовки. С началом сезонных соревнований дом культуры должен каждую неделю посылать в другой район команду из 6-ти человек, в которую должен войти 1 спортсмен из первой секции, 2 спортсмена из второй и 3 спортсмена из третьей (каждый спортсмен участвует в соревнованиях в течение одного сезона один раз). Сколько всего спортсменов занимается в доме культуры, если известно, что число вариантов отбора команды на четвертой неделе равно  $k$ ?

#### Решение

Пусть в первой секции занимается  $n$  спортсменов. Тогда во второй будет  $n + 3$ , а в третьей  $n + 6$  спортсменов. На 4-й неделе команду можно составить из  $n - 1 \cdot 3 = n - 3$  участников первой секции,  $n + 3 - 2 \cdot 3 = n - 3$  участников второй секции и  $n + 6 - 3 \cdot 3 = n - 3$  участников третьей секции. Таким образом, число вариантов:

$$C_{n-3}^1 \cdot C_{n-3}^2 \cdot C_{n-3}^3 = \frac{(n-3)^3(n-4)^2(n-5)}{12}.$$

Перебирая натуральные значения  $n = 6, 7, \dots$ , найдём такое  $n$ , что:

$$k = \frac{(n-3)^3(n-4)^2(n-5)}{12}.$$

(Перебор существенно сократится, если рассмотреть разложение числа  $12k$  на простые множители.)

Всего спортсменов, занимающихся в доме культуры, равно  $n + (n + 3) + (n + 6) = 3n + 9$ .

#### Диапазоны

Величина	min	max	Шаг
$n$	8	24	1

**Ответ:**  $3n + 9$ , если  $k = \frac{(n-3)^3(n-4)^2(n-5)}{12}$ .

### Задача II.2.4.3. (30 баллов)

Темы: стереометрия.



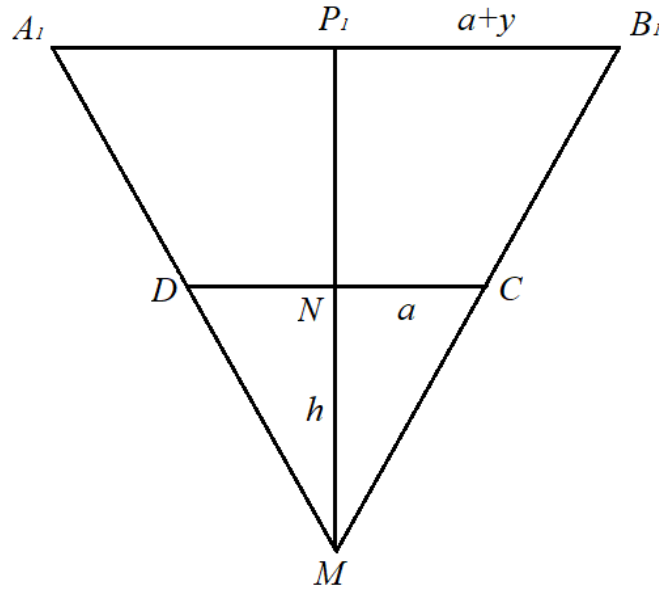
Отсюда:

$$h = \frac{2ra^2}{r^2 - a^2}.$$

Объём конуса равен:

$$V_1 = \frac{1}{3}\pi(a+x)^2(h+2r) = \frac{\pi}{3} \left(a + \frac{r^2 - a^2}{a}\right)^2 \left(\frac{2ra^2}{r^2 - a^2} + 2r\right) = \frac{2\pi r^7}{3a^2(r^2 - a^2)}.$$

После вынимания шара имеем следующую картину ( $A_1B_1$  — уровень воды):



Положим  $d = MP_1$  (высота нового конуса). Объём нового конуса равен:

$$V_2 = \frac{1}{3}\pi(a+y)^2d.$$

В то же время:

$$V_2 = V_1 - V_{\text{шара}} = V_1 - \frac{4}{3}\pi r^3.$$

Значит:

$$(a+y)^2d = \frac{3V_1}{\pi} - 4r^3.$$

Из подобия треугольников  $MNC$  и  $MP_1B_1$  находим:

$$\frac{a}{a+y} = \frac{h}{d},$$

поэтому:

$$a+y = \frac{ad}{h}.$$

Следовательно:

$$\left(\frac{ad}{h}\right)^2 d = \frac{3V_1}{\pi} - 4r^3.$$

Подставляя найденные ранее значения  $h$  и  $V_1$ , получаем:

$$d = \frac{2r}{r^2 - a^2} (r^6 - 2r^2a^2(r^2 - a^2))^{1/3}.$$

Тогда искомая величина  $NP_1$  равна:

$$NP_1 = d - h = \frac{2r}{r^2 - a^2} ((r^6 - 2r^4a^2 + 2r^2a^4)^{1/3} - a^2).$$

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$r$	10	20	0,5
$a$	4	8	0,5

Ответ:  $\frac{2r}{r^2 - a^2} ((r^6 - 2r^4a^2 + 2r^2a^4)^{1/3} - a^2)$ .

### Задача II.2.4.4. (25 баллов)

Темы: алгебра, задача на движение.

#### Условие

Два друга, Миша и Петя, хотят добраться из пункта  $A$  в пункт  $B$ . Расстояние между  $A$  и  $B$  равно  $S$  км. В распоряжении ребят есть мопед, передвигающийся со скоростью  $u$  км/ч. Отправляясь из  $A$  одновременно, Миша идёт пешком, а Петя едет на мопеде до встречи с Васей, идущим из  $B$  в  $A$ . Дальше Петя идет пешком, а Вася едет на мопеде до встречи с Мишей, после чего передает ему мопед, на котором Миша и добирается до  $B$ . За сколько часов до отправления Миши и Пети Вася должен выйти из пункта  $B$ , чтобы Миша и Петя добрались до  $B$  одновременно? Все три мальчика идут со скоростью  $v$  км/ч.

Записать ответ с точностью до 0,01.

#### Решение

Чтобы Миша и Петя прибыли в пункт  $B$  одновременно, они должны пройти пешком одно и то же расстояние  $x$  км и проехать на мопеде одно и то же расстояние  $S - x$  км. Тогда Вася до встречи с Петей должен пройти  $x$  км, и до встречи с Мишей проехать  $S - 2x$  км, а затем Миша проедет  $S - x$  км. Всего Вася и Миша проедут  $2S - 3x$  км. За это время Петя пройдет  $x$  км.

Тогда:

$$\frac{2S - 3x}{u} = \frac{x}{v},$$

следовательно:

$$x = \frac{2Sv}{u + 3v}.$$

Петя до встречи с Васей едет  $t_1 = \frac{S-x}{u}$  часов. Вася до встречи с Петей идёт  $t_2 = \frac{x}{v}$  часов. Тогда Вася должен выйти за  $t_2 - t_1$  часов до отправления Миши и Пети:

$$t_2 - t_1 = \frac{x}{v} - \frac{S-x}{u} = \frac{s(u-v)}{u(u+3v)}.$$

---

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$S$	15	30	5
$v$	5	7	1
$u$	14	35	1

Ответ:  $\frac{s(u-v)}{u(u+3v)}$ .

## Третья попытка. Задачи 8–9 класса

### Задача П.2.5.1. (20 баллов)

Темы: вероятность.

#### Условие

Участникам «математического марафона» предлагается для решения  $n$  задач,  $k$  из которых — задачи повышенной сложности. Задачи решаются по одной в случайном порядке. Участник получает следующую задачу лишь после того, как решил предыдущую. Решив задачу повышенной сложности, участник переходит на следующий уровень. Найти вероятность того, что для выхода на следующий уровень участнику придётся решить не менее пяти задач. Ответ округлить до сотых.

#### Решение

Если до выхода на следующий уровень участник решил не менее пяти задач, то первые четыре задачи не имели повышенной сложности, а сложность последующих задач не имеет значения. С учётом порядка всего имеется  $n(n-1)(n-2)(n-3)$  способов выбрать первые четыре задачи, и  $(n-k)(n-1-k)(n-2-k)(n-3-k)$  способов сделать первые четыре задачи простыми. Поэтому искомая вероятность равна:

$$p = \frac{(n-k)(n-1-k)(n-2-k)(n-3-k)}{n(n-1)(n-2)(n-3)}.$$

Погрешность 1%.

### Диапазоны

Величина	min	max	Шаг
$n$	10	15	1
$k$	3	6	1

Ответ:  $\frac{(n-k)(n-1-k)(n-2-k)(n-3-k)}{n(n-1)(n-2)(n-3)}$ .

---

### Задача II.2.5.2. (25 баллов)

Темы: комбинаторика.

#### Условие

В танцевальном кружке занимаются  $m$  девочек и  $n$  мальчиков, которые показывают примерно одинаковые результаты. В отборочных соревнованиях от кружка должны выступать  $k$  пар (каждой паре присваивается номер). Сколько существует вариантов составить пары?

#### Решение

Всего существует  $C_m^k$  способов выбрать группу девочек и  $C_n^k$  способов выбрать группу мальчиков, которые будут участвовать в соревнованиях. Далее каждому мальчику определим в пару девочку, это можно сделать  $k!$  способами. Наконец, нужно присвоить номера парам. Для этого есть  $k!$  способов. Итого имеем:

$$C_m^k \cdot C_n^k \cdot k! \cdot k! = \frac{m!n!}{(m-k)!(n-k)!}$$

способов составить пары.

#### Диапазоны

Величина	min	max	Шаг
$n$	7	10	1
$m$	6	7	1
$k$	3	5	1

Ответ:  $\frac{m!n!}{(m-k)!(n-k)!}$ .

### Задача II.2.5.3. (30 баллов)

Темы: планиметрия, площадь.

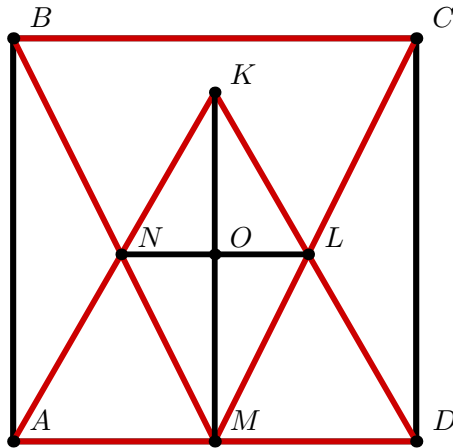
#### Условие

Точка  $K$  лежит внутри квадрата  $ABCD$  со стороной  $a$  так, что треугольник  $AKD$  равносторонний. Точка  $M$  делит отрезок  $AD$  пополам. Найти площадь фигуры, образованной пересечением треугольников  $AKD$  и  $BCM$ .

Ввести ответ с точностью до 0,01.



*Решение*



Пусть  $N = BM \cap KA$ ,  $L = CM \cap KD$ ,  $O = NL \cap KM$ . Ясно, что  $KM \perp NL$ . Поэтому площадь фигуры  $NKLM$

$$S = S_{NKL} + S_{NML} = \frac{1}{2}KO \cdot NL + \frac{1}{2}OM \cdot NL = \frac{1}{2}KM \cdot NL.$$

Пусть  $b = a/2$ . По теореме Пифагора из треугольника  $MKD$  находим:

$$KM = \sqrt{(2b)^2 - b^2} = b\sqrt{3}.$$

Треугольники  $ABN$  и  $MNK$  подобны с коэффициентом подобия:

$$k = \frac{2b}{b\sqrt{3}} = \frac{2}{\sqrt{3}}.$$

Следовательно,  $AN = k \cdot NK$ . Поэтому:

$$2b = AK = AN + NK = k \cdot NK + NK = (k + 1)NK.$$

Отсюда

$$NK = \frac{2b\sqrt{3}}{2 + \sqrt{3}}.$$

Треугольник  $NKL$  равносторонний, значит,  $NL = NK$ . Имеем:

$$S = \frac{1}{2}KM \cdot NL = \frac{1}{2} \cdot b\sqrt{3} \cdot \frac{2b\sqrt{3}}{2 + \sqrt{3}} = \frac{3b^2}{2 + \sqrt{3}} = \frac{3a^2}{4(2 + \sqrt{3})}.$$

Погрешность 1%.

*Диапазоны*

Величина	min	max	Шаг
$a$	2	40	4

Ответ:  $\frac{3a^2}{4(2 + \sqrt{3})}$ .

### Задача П.2.5.4. (25 баллов)

Темы: алгебра, задача на работу.

#### Условие

Петя, Вася и Иван участвуют в хакатоне по программированию. Если Петя и Вася работают вместе, то их производительности увеличиваются на 10% и 20% соответственно, и они могут написать программу, решающую предложенную задачу, за  $n$  часов. Иван же предпочитает индивидуальную работу, поэтому его производительность при совместной работе с Петей падает на 5%, а производительность Пети при таком взаимодействии возрастает на 6%. При совместной работе Ивана и Васи производительность Ивана падает на 15%, а производительность Васи возрастает на 14%. Петя и Иван, работая совместно, могут написать ту же программу за  $m$  часов, а Вася и Иван — за  $k$  часов. Сколько часов потратят все три мальчика, работая совместно, на написание этой программы, если при таком варианте сотрудничества производительность Пети возрастёт на 8%, Васи — на 17%, а Ивана — уменьшится на 10%? Ответ округлить до сотых.

#### Решение

Пусть  $x$  — производительность Пети,  $y$  — производительность Васи,  $z$  — производительность Ивана. Принимая за единицу объём работы, необходимый для решения задачи хакатона, получаем систему уравнений:

$$\begin{cases} 1,1x + 1,2y = \frac{1}{n}, \\ 1,06x + 0,95z = \frac{1}{m}, \\ 1,14y + 0,85z = \frac{1}{k}. \end{cases}$$

Складывая все уравнения, имеем:

$$2,16x + 2,34y + 1,8z = \frac{1}{n} + \frac{1}{m} + \frac{1}{k}.$$

Искомое время:

$$t = \frac{1}{1,08x + 1,17y + 0,9z} = \frac{2}{1/n + 1/m + 1/k}.$$

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$n$	4	7	1
$m$	5	8	1
$k$	6	9	1

Ответ:  $\frac{2}{1/n + 1/m + 1/k}$ .

---

## Третья попытка. Задачи 10–11 класса

### Задача П.2.6.1. (25 баллов)

Темы: вероятность, формула Байеса.

#### Условие

Участникам многопрофильной командной олимпиады предлагаются задачи по трём предметам: математика, информатика и физика. Модуль по математике содержит  $m$  задач,  $k$  из которых повышенной сложности. Модуль по информатике —  $n$  задач, среди которых  $l$  повышенной сложности. Модуль по физике —  $p$  задач, среди которых  $r$  повышенной сложности. Команды получают по три задачи, выбирая из каждого модуля случайно одну задачу. Участники команды  $\pi$  полученные задачи распределили между собой с помощью жеребьёвки, при этом оказалось, что Андрею выпала задача повышенной сложности. Определить вероятность того, что Андрей решает задачу по математике. Ответ округлить до сотых.

#### Решение

Пусть событие  $A$  — Андрею выпала задача повышенной сложности. Определим гипотезы, образующие полную группу событий:  $H_1$  — Андрею выпала задача по математике,  $H_2$  — по информатике,  $H_3$  — по физике. Тогда при  $i = 1, 2, 3$ :

$$P(H_i) = \frac{1}{3}, \quad P(A|H_1) = \frac{k}{m}, \quad P(A|H_2) = \frac{l}{n}, \quad P(A|H_3) = \frac{r}{p}.$$

По формуле Байеса искомая вероятность равна:

$$P(H_1|A) = \frac{P(H_1)P(A|H_1)}{\sum_{i=1}^3 P(H_i)P(A|H_i)} = \frac{k/m}{k/m + l/n + r/p}$$

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$m$	8	10	1
$k$	4	5	1
$n$	6	8	1
$l$	3	4	1
$p$	7	9	1
$r$	3	4	1

Ответ:  $\frac{k/m}{k/m + l/n + r/p}$ .

### Задача II.2.6.2. (20 баллов)

Темы: комбинаторика.

#### Условие

Сколько различных комбинаций можно составить из  $n$  различных цифр, наименьшая из которых в комбинации повторяется ровно  $k$  раз, а остальные по одному разу, и первых  $m$  букв латинского алфавита, если все цифры должны стоять рядом, а буква «с» должна стоять непосредственно перед буквой «а»?

#### Решение

Последовательно выбираем:

1. позиции для повторяющейся цифры относительно других цифр —  $C_{n-1+k}^k$  способов;
2. порядок неповторяющихся цифр —  $(n-1)!$  способов;
3. порядок букв —  $(m-1)!$  способов (поскольку буква «с» стоит непосредственно перед буквой «а», то можно считать, что у нас имеется  $m-1$  буква);
4. положение группы цифр относительно букв —  $m$  способов.

Таким образом, всего имеется:

$$C_{n-1+k}^k (n-1)! (m-1)! m = \frac{(n-1+k)! m!}{k!}$$

#### Диапазоны

Величина	min	max	Шаг
$n$	3	5	1
$k$	3	5	1
$m$	5	9	1

Ответ:  $\frac{(n-1+k)! m!}{k!}$ .

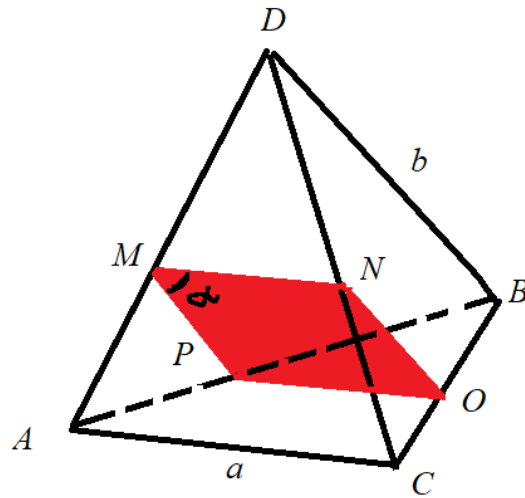
### Задача II.2.6.3. (30 баллов)

Темы: стереометрия.

#### Условие

В тетраэдре  $ABCD$  среди сечений, параллельных рёбрам  $AC$  и  $BD$  одновременно, выбрано сечение с наибольшей площадью. Длина ребра  $AC$  равна  $a$ , длина ребра  $BD$  равна  $b$ . Найти сумму квадратов диагоналей этого сечения с точностью до 0,1.

*Решение*



Пусть  $PMNO$  — секущая плоскость. Имеем:

1.  $AC, PO \subset ABC, AC \parallel MNP \Rightarrow PO \parallel AC$ .
2.  $BD, NO \subset DBC, BD \parallel MNP \Rightarrow NO \parallel BD$ .
3. Аналогично,  $MN \parallel AC, MP \parallel BD$ .
4. Из пунктов 1, 2 и 3 следует, что противоположные стороны четырёхугольника  $MNOP$  параллельны, а значит,  $MNOP$  — параллелограмм.
5. Треугольники  $PBO$  и  $ABC$  подобны с коэффициентом подобия  $k$ . Тогда  $PO = MN = ka$ .
6. Треугольники  $APM$  и  $ABD$  подобны с коэффициентом подобия  $1 - k$ . Тогда  $MP = NO = (1 - k)b$ .
7. Угол между прямыми  $AC$  и  $BD$  равен углу между прямыми  $MP$  и  $MN$ , обозначим его через  $\alpha$ . Площадь сечения равна:

$$S = MP \cdot MN \cdot \sin \alpha = ab \sin \alpha \cdot k(1 - k).$$

Отсюда получаем, что площадь сечения максимальна при  $k = 1/2$ .

8. Стороны сечения равны  $PO = MN = \frac{a}{2}, MP = NO = \frac{b}{2}$ .
9. Сумма квадратов диагоналей параллелограмма равна сумме квадратов всех его сторон:

$$MO^2 + NP^2 = 2(MP^2 + MN^2) = \frac{a^2 + b^2}{2}.$$

Погрешность 1%.

*Диапазоны*

Величина	min	max	Шаг
$a$	4	10	0,5
$b$	11	20	0,5

Ответ:  $\frac{a^2 + b^2}{2}$ .

### Задача П.2.6.4. (25 баллов)

Темы: алгебра, функциональное уравнение.

#### Условие

Найти значение  $f(1)$  для функции  $f$ , удовлетворяющей при всех  $x \neq a$  уравнению:

$$f(x) + k \cdot f\left(\frac{ax+b}{x-a}\right) = nx.$$

Записать ответ с точностью до 0,01.

#### Решение

Если  $y = \frac{ax+b}{x-a}$ , то  $x = \frac{ay+b}{y-a}$ . Тогда, подставляя вместо  $x$  в уравнении значение  $y$ , находим:

$$f\left(\frac{ax+b}{x-a}\right) + kf(x) = n \cdot \frac{ax+b}{x-a}.$$

Положим  $A = f(x)$ ,  $B = f\left(\frac{ax+b}{x-a}\right)$ . Тогда имеем систему уравнений:

$$\begin{cases} A + kB = nx, \\ B + kA = n \cdot \frac{ax+b}{x-a}. \end{cases}$$

Решая её относительно переменных  $A$  и  $B$ , находим:

$$A = \frac{n(-x^2 + ax(k+1) + kb)}{(x-a)(k^2-1)}.$$

Следовательно:

$$f(1) = \frac{n(a(k+1) + kb - 1)}{(1-a)(k^2-1)}.$$

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$a$	10	40	10
$b$	1	10	1
$k$	0,1	0,9	0,1
$n$	5	9	1

Ответ:  $\frac{n(a(k+1) + kb - 1)}{(1-a)(k^2-1)}$ .

---

## Четвертая попытка. Задачи 8–9 класса

### Задача П.2.7.1. (25 баллов)

Темы: вероятность.

#### Условие

Участникам олимпиады предлагается для решения  $n$  задач. Вася может решить каждую из задач с вероятностью  $p$ . Найти вероятность того, что Вася решит не более одной задачи. Ответ округлить до сотых.

#### Решение

Вероятность того, что Вася не решит ни одной задачи:

$$p_0 = (1 - p)^n.$$

Вероятность того, что Вася решит ровно одну задачу:

$$p_1 = np(1 - p)^{n-1}.$$

Значит, вероятность того, что Вася решит не более одной задачи:

$$p_{0,1} = p_0 + p_1 = (1 - p)^n + np(1 - p)^{n-1}.$$

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$n$	10	15	1
$p$	0,1	0,4	0,1

Ответ:  $(1 - p)^n + np(1 - p)^{n-1}$ .

### Задача П.2.7.2. (20 баллов)

Темы: комбинаторика.

#### Условие

На спортивные соревнования из другого района приехало  $n$  школьников. В гостинице на этот момент оказались свободными только один двухместный номер, один трёхместный номер и один четырёхместный номер. Остальных школьников решили разместить в спортзале местной школы. Руководитель должен составить список на расселение. Сколько возможно вариантов такого списка, если порядок записанных в один номер гостиницы (или в спортзал) не важен?

---

### Решение

Последовательно выбираем спортсменов:

1. в двухместный номер —  $C_n^2$  способов;
2. в трёхместный номер —  $C_{n-2}^3$  способов;
3. в четырёхместный номер —  $C_{n-5}^4$  способов;
4. в спортзал разместятся все остальные спортсмены — 1 способ.

Таким образом, общее число вариантов:

$$C_n^2 C_{n-2}^3 C_{n-5}^4 = \frac{n!}{2!(n-2)!} \frac{(n-2)!}{3!(n-5)!} \frac{(n-5)!}{4!(n-9)!} = \frac{n!}{288 \cdot (n-9)!}.$$

### Диапазоны

Величина	min	max	Шаг
$n$	14	25	1

Ответ:  $\frac{n!}{288 \cdot (n-9)!}$ .

### Задача II.2.7.3. (25 баллов)

Темы: планиметрия.

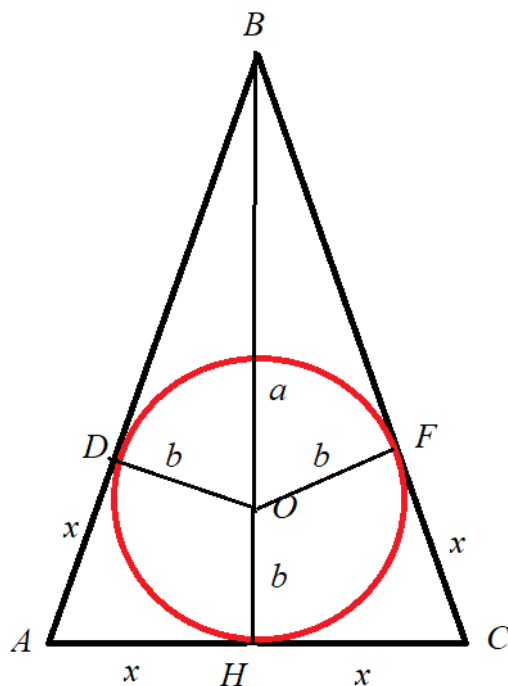
#### Условие

В равнобедренном треугольнике  $ABC$  с основанием  $AC$  вписана окружность с центром в точке  $O$ . Точка  $O$  делит высоту  $BH$  на отрезки длиной  $a$  и  $b$ , считая от точки  $B$ . Найти периметр треугольника  $ABC$ .

Ввести ответ с точностью до 0,01.



*Решение*



Пусть  $D$  и  $F$  — точки касания окружности с боковыми сторонами треугольника,  $x$  — длина  $AH$ . Тогда  $AD = AH = CH = CF = x$ .

Из треугольника  $OBD$  по теореме Пифагора получаем  $BD^2 = a^2 - b^2$ . Следовательно, в треугольнике  $ABH$ :  $(\sqrt{a^2 - b^2} + x)^2 = (a + b)^2 + x^2$ .

Отсюда  $x = b \frac{a+b}{\sqrt{a^2 - b^2}}$ .

Периметр треугольника  $ABC$  равен:

$$4x + 2\sqrt{a^2 - b^2} = \frac{2(a + b)^2}{\sqrt{a^2 - b^2}}.$$

Погрешность 1%.

*Диапазоны*

Величина	min	max	Шаг
$a$	10	20	1
$b$	2	9	1

Ответ:  $\frac{2(a + b)^2}{\sqrt{a^2 - b^2}}$ .

**Задача II.2.7.4. (30 баллов)**

Темы: алгебра, поиск шаблона.

### Условие

Петя приехал на курорт на  $k$  дней, имея на банковской карте целое число тысяч рублей. Каждый день он тратил половину имеющихся денег и еще 500 рублей, причём остаток на его счете в конце дня всегда делился на тысячу нацело. В последний день были потрачены все деньги. Сколько изначально рублей было у Пети?

### Решение

Пусть у Пети  $n$  тыс. руб.

В первый день он потратил  $\frac{n}{2} + \frac{1}{2} = \frac{n+1}{2}$  тыс. руб. У него осталось  $n - \frac{n+1}{2} = \frac{n-1}{2}$  тыс. руб.

Во второй день он потратил  $\frac{n-1}{4} + \frac{1}{2} = \frac{n+1}{4}$  тыс. руб. У него осталось  $\frac{n-1}{2} - \frac{n+1}{4} = \frac{n-3}{4}$  тыс. руб.

В третий день он потратил  $\frac{n-3}{8} + \frac{1}{2} = \frac{n+1}{8}$  тыс. руб. У него осталось  $\frac{n-3}{4} - \frac{n+1}{8} = \frac{n-7}{8}$  тыс. руб.

И так далее: в  $j$ -ый день он потратит  $\frac{n+1}{2^j}$  тыс. руб.

В последний день он потратит одну тыс. руб., то есть:

$$\frac{n+1}{2^k} = 1.$$

Поэтому  $n = 2^k - 1$ . Тогда всего рублей  $1000(2^k - 1)$ .

### Диапазоны

Величина	min	max	Шаг
$k$	6	15	1

Ответ:  $n = 1000(2^k - 1)$ .

## Четвертая попытка. Задачи 10–11 класса

### Задача П.2.8.1. (25 баллов)

Темы: теория вероятностей.

### Условие

В районной «математической регате» участвует  $2^m$  команд, причём физико-математический лицей выставил две команды. По правилам соревнований в каждом туре команды случайным образом разбиваются на игровые пары. В каждой паре определяется проигравшая команда, которая выбывает из турнира, и победившая, которая выходит в следующий тур. В каждой встрече вероятность выигрыша и поражения для любой команды равна 0,5. Какова вероятность того, что команды физико-математического лицея сыграют друг с другом в  $k$ -м туре?

Ответ округлить до тысячных.

## Решение

Пусть событие  $A_k$  — команды физико-математического лица встретились на  $k$ -м туре; событие  $B_k$  — команды физико-математического лица прошли в  $k$ -й тур. Имеем следующую цепочку вложений:

$$\bar{A}_1 \supset B_2 \bar{A}_2 \supset B_3 \bar{A}_3 \supset \dots \supset B_{k-1} \bar{A}_{k-1} \supset B_k \supset A_k,$$

поэтому

$$A_k = \bar{A}_1 B_2 \bar{A}_2 B_3 \cdot \dots \cdot \bar{A}_{k-1} B_k A_k.$$

По формуле для вероятности произведения получаем:

$$P(A_k) = P(\bar{A}_1)P(B_2|\bar{A}_1)P(\bar{A}_2|\bar{A}_1 B_2) \cdot \dots \cdot P(A_k|\bar{A}_1 B_2 \bar{A}_2 B_3 \cdot \dots \cdot \bar{A}_{k-1} B_k)$$

Последовательно находим:

$$P(\bar{A}_1) = 1 - \frac{2^m/2}{C_{2^m}^2} = \frac{2(2^{m-1} - 1)}{2^m - 1},$$

$$P(B_2|\bar{A}_1) = 0,5 \cdot 0,5 = \frac{1}{4},$$

$$P(\bar{A}_2|\bar{A}_1 B_2) = \frac{2(2^{m-2} - 1)}{2^{m-1} - 1} \quad (\text{вычисляется аналогично } P(\bar{A}_1)),$$

$$P(B_3|\bar{A}_1 B_2 \bar{A}_2) = 0,5 \cdot 0,5 = \frac{1}{4},$$

...

$$P(\bar{A}_{k-1}|\bar{A}_1 B_2 \bar{A}_2 B_3 \cdot \dots \cdot B_{k-1}) = \frac{2(2^{m-(k-1)} - 1)}{2^{m-(k-1)+1} - 1},$$

$$P(B_k|\bar{A}_1 B_2 \bar{A}_2 B_3 \cdot \dots \cdot B_{k-1} \bar{A}_{k-1}) = 0,5 \cdot 0,5 = \frac{1}{4},$$

$$P(A_k|\bar{A}_1 B_2 \bar{A}_2 B_3 \cdot \dots \cdot \bar{A}_{k-1} B_k) = \frac{2^{m+1-k}/2}{C_{2^{m+1-k}}^2} = \frac{1}{2^{m+1-k} - 1}$$

Перемножая полученные значения, получаем:

$$\begin{aligned} P(A_k) &= \frac{2(2^{m-1} - 1)}{2^m - 1} \cdot \frac{1}{4} \cdot \frac{2(2^{m-2} - 1)}{2^{m-1} - 1} \cdot \frac{1}{4} \cdot \dots \cdot \frac{2(2^{m-(k-1)} - 1)}{2^{m-(k-1)+1} - 1} \cdot \frac{1}{4} \cdot \frac{1}{2^{m+1-k} - 1} \\ &= \frac{1}{2^{k-1}(2^m - 1)}. \end{aligned}$$

Погрешность 0,1%.

## Диапазоны

Величина	min	max	Шаг
$m$	5	7	1
$k$	3	5	1

Ответ:  $\frac{1}{2^{k-1}(2^m - 1)}$ .

### Задача II.2.8.2. (20 баллов)

Темы: комбинаторика, сочетания с повторениями.

#### Условие

Команде из  $k$  горнолыжников перед соревнованием на новой трассе предоставляется возможность совершить  $n$  тренировочных спусков. Тренер должен указать, сколько тренировочных спусков предоставляется каждому спортсмену его команды. Сколько существует вариантов такого распределения, если известно, что каждый спортсмен должен совершить не менее  $m$  тренировочных спусков?

#### Решение

Предоставив  $m$  тренировочных спусков каждому спортсмену, останется распределить  $n - km$  спусков. Это схема выбора  $j = n - km$  предметов из  $l = k$  без упорядочения и с повторениями. Количество всевозможных таких наборов — число сочетаний с повторениями:

$$\bar{C}_l^j = C_{l+j-1}^j.$$

Подставляя значения  $l$  и  $j$ , находим:

$$\bar{C}_k^{n-km} = C_{k+n-km-1}^{n-km} = \frac{(k+n-km-1)!}{(n-km)!(k-1)!}.$$

#### Диапазоны

Величина	min	max	Шаг
$n$	60	100	1
$k$	4	7	1
$m$	5	7	1

Ответ:  $\frac{(k+n-km-1)!}{(n-km)!(k-1)!}$ .

### Задача II.2.8.3. (30 баллов)

Темы: стереометрия.

#### Условие

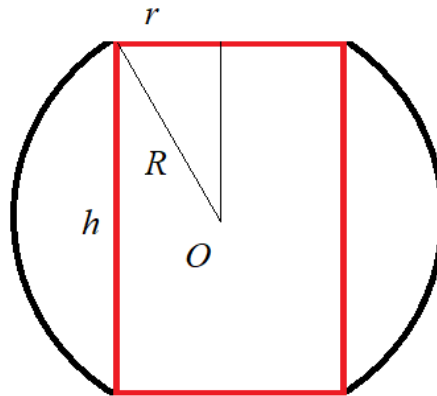
В деревянном шаре радиуса  $R$  плотности  $\rho_1$  просверлили сквозное цилиндрическое отверстие (ось цилиндра проходит через центр шара) и заполнили образовавшуюся полость металлическим сплавом плотности  $\rho_2$ . Каков наибольший радиус этого отверстия, при котором полученное тело не утонет в жидкости плотности  $\rho_3$ ?

Примечание: плотность тела — это отношение массы тела к его объёму. Из закона Архимеда следует, что тело не тонет, если его плотность не превосходит плотности жидкости.

Записать ответ с точностью до 0,01.

## Решение

Пусть  $r$  — искомый радиус цилиндра,  $h$  — высота цилиндра. Изобразим сечение полученного тела плоскостью, проходящей через ось цилиндра.



Плотность полученного тела, имеющего массу  $M$  и объём  $V$ , равна:

$$\rho = \frac{M}{V} = \frac{V_1\rho_1 + V_2\rho_2}{V_1 + V_2},$$

где  $V_1$  — объём деревянной части,  $V_2$  — металлической.

Объём цилиндра равен:

$$V_2 = h \cdot \pi r^2.$$

Для нахождения объёма деревянной части вычтем из шарового слоя объём цилиндра:

$$V_1 = V_{\text{шар. сл.}} - V_2 = \pi h(r^2 + h^2/6) - h\pi r^2 = \frac{\pi h^3}{6}.$$

Условие, при котором тело не тонет, имеет вид  $\rho \leq \rho_3$ , то есть:

$$\frac{\frac{\pi h^3}{6}\rho_1 + h\pi r^2\rho_2}{\pi h(r^2 + h^2/6)} \leq \rho_3,$$

что равносильно:

$$r^2(\rho_2 - \rho_3) \leq \frac{h^2}{6}(\rho_3 - \rho_1).$$

Подставляя  $h^2 = 4(R^2 - r^2)$  (из теоремы Пифагора), после несложных преобразований приходим к неравенству:

$$r \leq R \sqrt{\frac{2(\rho_3 - \rho_1)}{3\rho_2 - \rho_3 - 2\rho_1}}.$$

Погрешность 1%.

## Диапазоны

Величина	min	max	Шаг
$R$	2	10	1
$\rho_1$	0,5	0,9	0,1
$\rho_2$	7	12	1
$\rho_3$	1	4	0,5

---

Ответ:  $R\sqrt{\frac{2(\rho_3 - \rho_1)}{3\rho_2 - \rho_3 - 2\rho_1}}$ .

### Задача II.2.8.4. (25 баллов)

Темы: алгебра, функциональное уравнение.

#### Условие

Дима и Лёша хотят попасть из пункта  $A$  в пункт  $B$ . Расстояние между  $A$  и  $B$  равно  $S$  км. В распоряжении ребят есть скейтборд, скорость передвижения на котором равна  $u$  км/ч. Отправляясь одновременно, Дима идёт пешком, а Лёша едет на скейтборде до встречи с Пашей, который вышел из  $B$  в  $A$  за  $t$  часов до отправления Димы и Лёши. Дальше Лёша идёт пешком, а Паша едет на скейтборде до встречи с Димой, после чего передаёт ему скейтборд, на котором тот и добирается до  $B$ . Известно, что Дима и Лёша добрались до  $B$  одновременно. Чему равна скорость ходьбы мальчиков (в км/ч), если известно, что она у всех одинаковая?

Записать ответ с точностью до 0,01.

#### Решение

Пусть скорость мальчиков равна  $v$  км/ч. Чтобы Дима и Лёша прибыли в пункт  $B$  одновременно, они должны пройти пешком одно и то же расстояние  $x$  км и проехать на скейтборде одно и то же расстояние  $S - x$  км. Тогда Паша до встречи с Лёшей должен пройти  $x$  км, и до встречи с Димой проехать  $S - 2x$  км, а затем Дима проедет  $S - x$  км. Всего Паша и Дима проедут  $2S - 3x$  км. За это время Лёша пройдёт  $x$  км.

Тогда:

$$\frac{2S - 3x}{u} = \frac{x}{v}.$$

Лёша до встречи с Пашей едет  $\frac{S-x}{u}$  часов. Паша до встречи с Лёшей идёт  $\frac{x}{v}$  часов.

Тогда:

$$\frac{x}{v} - \frac{S-x}{u} = t.$$

Из системы:

$$\begin{cases} \frac{2S-3x}{u} = \frac{x}{v}, \\ \frac{x}{v} - \frac{S-x}{u} = t \end{cases}$$

находим  $v = \frac{u(S - ut)}{S + 3ut}$ .

Погрешность 1%.

#### Диапазоны

Величина	min	max	Шаг
$S$	30	40	5
$u$	10	12	1
$t$	0,25	0,75	0,25

Ответ:  $\frac{u(S - ut)}{S + 3ut}$ .

# Инженерный тур

## *Введение*

Задачи первого этапа инженерного тура по профилю «Технологии дополненной реальности» направлены на инициализацию базовых понятий и отработку уже имеющихся навыков по математике и программированию. Эти задания не преследуют цели обнаружения какого-либо уровня владения конкретными инструментами реализации проектов дополненной реальности. С их помощью мы хотим найти ребят, у которых есть интерес к профилю и соответствующие базовые навыки для приобретения новых компетенций, необходимых при разработке финального проекта.

## *Задача П.3.1. Континуум реальное-виртуальное (2 балла)*

*Темы: дополненная реальность, континуум реальное виртуальное, виртуальная реальность, дополненная виртуальность.*

### *Условие*

В 1994 году Пол Милгрэм и Фумио Кисино описали континуум «реальность — виртуальность» — пространство между объективной реальностью и виртуальным миром, который полностью смоделирован на компьютере.

Между реальным и виртуальным миром находится континуум реальность–виртуальность, который охватывает все возможные вариации и составы реальных и виртуальных объектов.

Отличительными характеристиками, по которым выделяются разные типы «реальностей», является уровень или глубина погружения в виртуальное пространство. Погружение в виртуальность — это восприятие физического присутствия в нефизическом мире. Восприятие создается путем окружения пользователем системы изображениями, звуками или другими стимулами, которые обеспечивают увлекательную общую среду. Чем глубже погружение, тем выше такая характеристика системы, как иммерсивность (способ восприятия, создающий эффект погружения в искусственно созданную среду).

Ниже приведен нумерованный список из определений цифровых реальностей. Определите, о каких цифровых реальностях идет речь. В ответ запишите номера цифровых реальностей в строку в том порядке, в котором они расположены на континууме реальное-виртуальное.

1. Созданный техническими средствами мир, передаваемый человеку через его ощущения: зрение, слух, осязание и другие, технология имитирует как воздействие, так и реакции на воздействие.
2. Интерактивный опыт, который сочетает в себе реальный мир и компьютерный контент. Контент может охватывать несколько сенсорных модальностей, включая визуальную, слуховую, тактильную, соматосенсорную и обонятельную.
3. Технология, которая способна соединить отдаленных друг от друга участников общения в одном виртуальном пространстве, симулируя реальную встречу

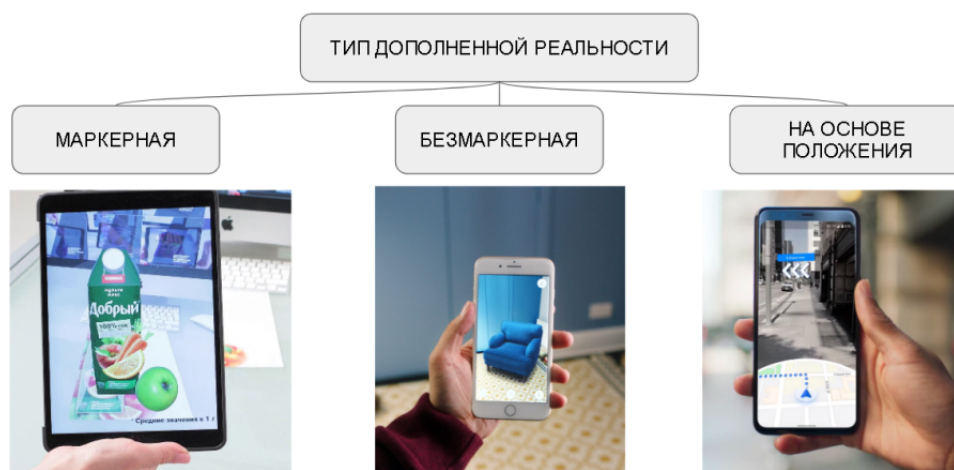
Ответ: 2, 3, 1.

### Задача II.3.2. Типы дополненной реальности (2 балла)

Темы: маркерная дополненная реальность, безмаркерная дополненная реальность, дополненная реальность на основе местоположения.

#### Условие

Всего существует три типа приложений с дополненной реальностью: маркерная, безмаркерная и дополненная реальность на основе местоположения.



Типы дополненной реальности используют разнообразные механики. Сопоставьте описание механики с ее названием.

1. Механика распознавания изображений или таргетинг изображения	А. Image tracking
2. Механика обнаружения объектов в пространстве 3D	Б. Object tracking
3. Механика отслеживания лица в режиме реального времени в 3D	В. Face tracking
4. Механика, которая позволяет приложениям отображать среду и отслеживать свои собственные движения в ней	Г. SLAM
5. Самая простая форма механики SLAM, которая обеспечивает понимание приложением пола, барьеров и стен	Д. Plane detection
6. AR-приложение позиционирует контент дополненной реальности на конкретном местоположении, определяя его с помощью GPS-координат	Е. Location-based Augmented Reality

Ответ: 1 – А, 2 – Б, 3 – В, 4 – Г, 5 – Д, 6 – Е.

### Задача II.3.3. Триггер (2 балла)

Темы: дополненная реальность, триггер, оверлей.

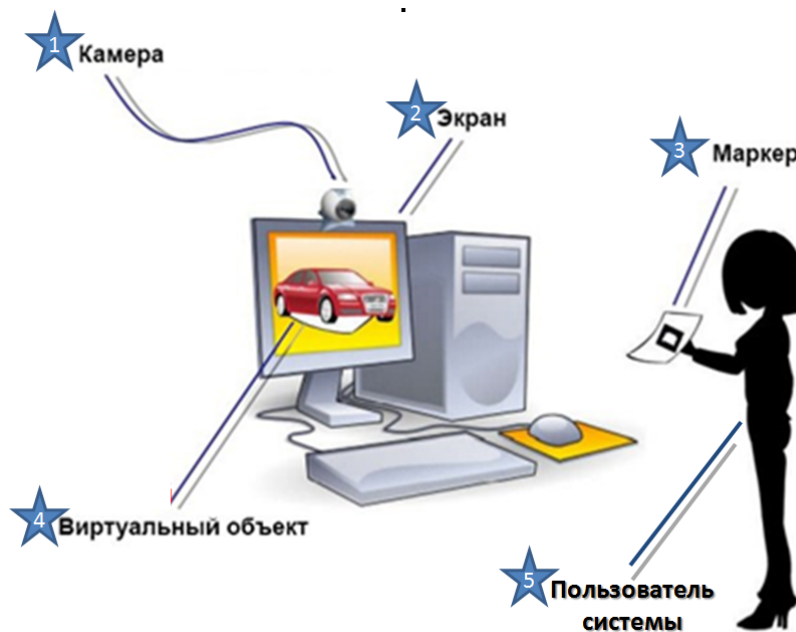


## Условие

Дополненная реальность (AR) — это интерактивный опыт, который сочетает в себе реальный мир и компьютерный контент. Контент может охватывать несколько сенсорных модальностей, включая визуальную, слуховую, тактильную, соматосенсорную и обонятельную.

Разберемся, как работает технология дополненной реальности. Камера захватывает и сканирует реальный объект или местоположение пользователя, производится анализ изображения, или данных с датчиков (например, GPS), система ищет совпадение с базой данных, и после найденного совпадения на экран выводится виртуальный объект. Объект реального мира, являющийся поводом поставления в видеопоток камеры устройства пользователя дополнительной информации в виде виртуальных объектов, — это триггер. Информация, добавляемая в видеопоток камеры устройства пользователя, при считывании маркера, распознавании 3D-объекта, при нахождении в определенной точке пространства — это оверлей.

Рассмотрите приведенную ниже схему, укажите последовательно номер объекта, который является триггером, а затем номер, закрепленный за оверлеем. В ответе должно получиться двузначное число, первая цифра которого номер триггера, вторая — оверлея.



Ответ: 34.

## Задача II.3.4. Оверлей (16 баллов)

Темы: оверлей, цифровой аватар, трехмерная модель, Blender.

## Условие

Оверлеи бывают разные. Очень популярны трехмерные модели — цифровые аватары объектов реального мира.

Мы будем делать вазу или кувшин (возможно, там мы спрячем клад).

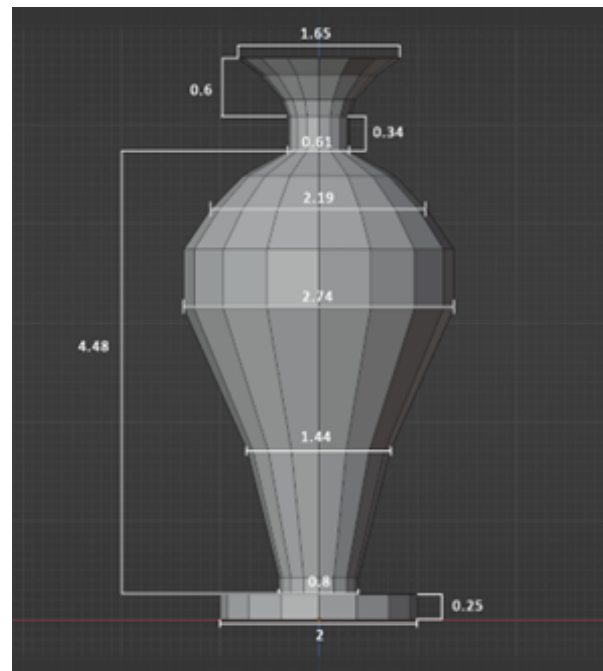
Внизу на рисунке слева представлен внешний вид вазы, а справа дан чертеж, которого необходимо придерживаться при создании оверлея. На нем слева указана высота элементов, справа — ширина (длина диагонали). Ваша задача — воссоздать вазу как можно ближе к чертежу.

В приложении к заданию вы найдете blender-файл, а также чертеж вазы: [https://disk.yandex.ru/d/BKW-7tYZf\\_ia6Q](https://disk.yandex.ru/d/BKW-7tYZf_ia6Q).

Обратите внимание, что в файле имеется ряд настроек, которые нельзя менять:

1. положение камеры;
2. положение источников освещения;
3. настройки рендера.

Создайте свою вазу на основе чертежа в этом blender-файле, и в отклик прикрепите рендер вашей работы в формате png. Предполагается, что основание вазы находится в точке  $(0, 0, 0)$  и отмечено на чертеже оранжевой точкой.



**Ответ:** модельное решение доступно по ссылке: <https://disk.yandex.ru/d/GA05i4hMqmmtWw>.

### *Код проверки решения*

```
from skimage.metrics import structural_similarity as compare_ssim
import cv2
if __name__ == "__main__":
    imageA = cv2.imread("путь к оригинальному изображению")
    imageB = cv2.imread("путь к изображению из отклика")

    grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
    grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)

    score = compare_ssim(grayA, grayB, full=True)[0] * 100
    if score > 95:
        print(True)
```

```
else:  
    print(False)
```

Где `score` — регулируемая обсуждаемая величина схожести.

### **Задача II.3.5. Клад (26 баллов)**

Темы: *дополненная реальность на основе местоположения, геопозиционирование, триангуляция.*

#### **Условие**

Использование систем дополненной реальности, которые ориентируются на сигналы систем геопозиционирования, позволяет разработчикам указывать реальные координаты, в которых будет отображаться AR-контент. Система определяет местоположение пользователя (отслеживая устройство, которое он использует) если пользователь находится рядом с указанной разработчиком точкой, то он сможет увидеть виртуальные объекты на экране своего смартфона. Примеры работы представлены на картинке ниже.



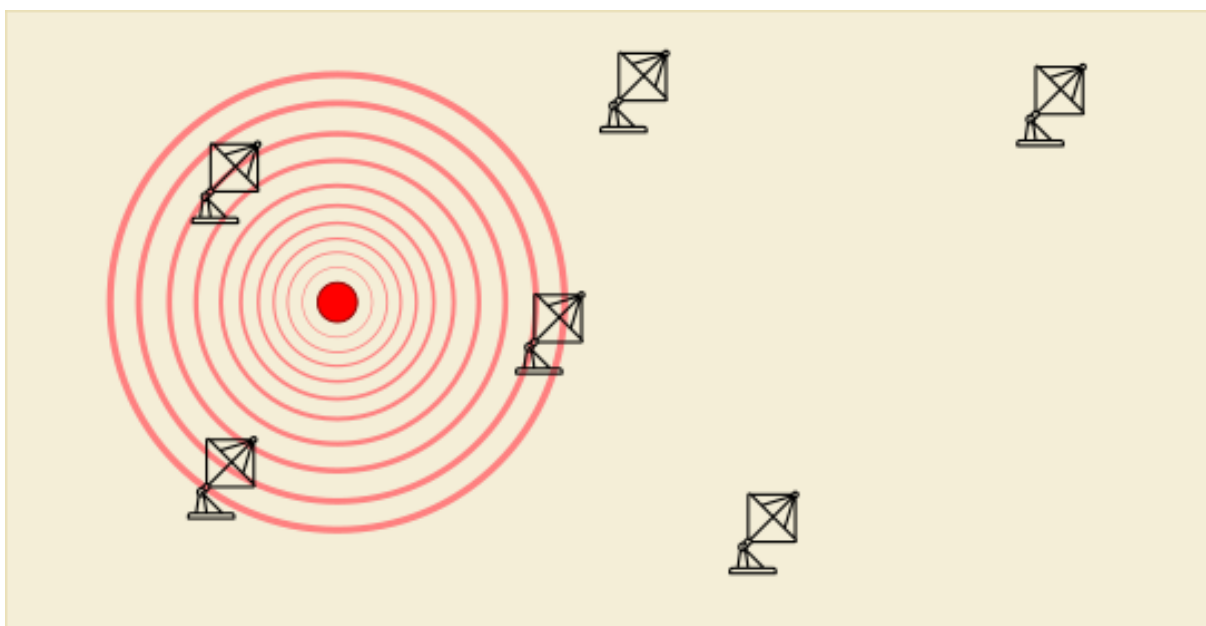
Рис. II.3.1. На смартфон пользователя пришло оповещение, что он подошел к нужной локации



Рис. II.3.2. Сканируя пространство, пользователь находит искомый виртуальный объект

Географическое местоположение любого GSM-телефона (а значит и пользователя) можно с достаточно большой точностью определить при помощи триангуляции по трем базовым станциям GPS. Описывают это обычно так: допустим, если можно стандартными средствами определить расстояние от базовой станции до телефона, то по расстояниям от трех базовых станций можно получить точные координаты аппарата, а по расстоянию от двух базовых станций — две точки, в одной из которых и будет находиться искомый телефон. Примером использования технологии дополненной реальности, основанной на определении местоположения пользователя, является игра Pokémon Go.

В нашем случае разработчики реализуют квест с внедрением AR-контента «Поиск клада». Для игроков составлена карта с предполагаемыми местами его расположения, а сам клад снабжен радиомаяком. На территории расположены станции слежения, которые измеряют расстояние пользователя до точки нахождения клада. На рисунке ниже приведен пример с меньшим количеством станций, красным обозначена точка клада.



Координаты всех станций известны и приведены ниже. Три ближайшие к месту нахождения радиомаяка станции сообщают вам расстояние до него. Также вам выдается список предполагаемых точек местонахождения клада. Используя метод триангуляции, выберите из всех предполагаемых точек ту, в которой спрятан клад.

Координаты станций слежения:

53 93  
145 58  
123 62  
65 96  
123 49  
67 90  
171 97  
186 89  
130 67  
56 94  
29 16  
34 65  
80 1  
4 11  
186 27  
48 95  
122 13  
194 20  
74 65  
124 13

### *Формат входных данных*

1-я строка: расстояния от клада до ближайших станций — три вещественных числа, разделенные пробелом.

---

Следующие 20 строк: предполагаемые координаты клада — по два числа, разделенные пробелом, каждая пара в отдельной строке.

### *Формат выходных данных*

Номер строки, содержащей координаты клада (одно число от 0 до 19).

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```
1  treasure_points_str = """172 18
2  154 62
3  10 84
4  107 31
5  41 97
6  184 56
7  81 38
8  118 33
9  146 70
10 196 100
11 106 96
12 77 77
13 130 54
14 80 80
15 60 66
16 155 58
17 137 17
18 162 33
19 199 41
20 73 5"""
21 treasure_points = [ tuple(map(int, s.split())) for s in
   ↪ treasure_points_str.split('\n') ]
22 stations_text = """53 93
23 145 58
24 123 62
25 65 96
26 123 49
27 67 90
28 171 97
29 186 89
30 130 67
31 56 94
32 29 16
33 34 65
34 80 1
35 4 11
36 186 27
37 48 95
38 122 13
39 194 20
40 74 65
41 124 13"""
42 stations = []
43 for s in stations_text.split('\n'):
44     stations.append(tuple(map(int, s.split())))
45 def treasure_point_index(dst, stations, points):
```

```

46     deltas = []
47     for p in points:
48         delta = 0
49         for d in dst:
50             delta += min([abs(distance(st, p) - d) for st in stations ] )
51         deltas.append(delta)
52         #print("%.2f" % delta)
53         min_index = deltas.index(min(deltas))
54     return str(min_index)
55
56 print(treasure_point_index((12.04, 16.28, 24.35), stations, treasure_points))

```

### ***Задача II.3.6. Траектория движения кубитоклобуса (26 баллов)***

*Темы: компьютерное зрение.*

#### ***Условие***

Если вам не удалось найти клад (кувшин с деньгами), то, скорее всего, его стащил кубитоклобус.

Когда-то мы решили, что одной из существей, населяющих фиджитал-мир являются кубитоклобусы. На самом деле мы их сами туда поселили. Эти милые обитатели цифровых глубин проще всего взаимодействуют с начинающими свое погружение разработчиками, поскольку имеют простые и понятные формы. Чтобы выследить кубитоклобуса, применяются методы компьютерного зрения — те самые, которые составляют основу оптической дополненной реальности.

Сейчас ведется наблюдение за объектом круглой формы. Необходимо определить траекторию его движения. Известно, что объект может смещаться либо влево на 1 пиксель, либо вправо на 1 пиксель, либо вверх на 1 пиксель, либо вниз на 1 пиксель.

#### ***Формат входных данных***

Ряд из  $N$  бинарных снимков размера  $64 \times 64$ .

#### ***Формат выходных данных***

Массив размера  $N - 1$  — траектория движения заданная векторами  $(0, 1)$ ,  $(0, -1)$ ,  $(1, 0)$ ,  $(-1, 0)$ .

#### ***Генерация кадров***

```

def generate_trajectory(R=5,N=10, H=64, W=64):
    directions = [(0,1), (0, -1), (1, 0), (-1,0)]
    trajectory = []
    images = []
    start_img = np.zeros((H,W))
    coords = (random.randint(R,H-R), random.randint(R,W-R))
    mask = draw_circle(start_img, R, coord=coords)
    start_img[mask] = 1
    images.append(start_img)

```

---

```

for i in range(N-1):
    dir_id = random.randint(0,3)
    new_direction = directions[dir_id]
    new_circle = np.zeros((H,W))
    if not (R < coords[0] + new_direction[0] < H-R):
        new_direction = (-new_direction[0],new_direction[1])

    if not (R < coords[1] + new_direction[1] < W-R):
        new_direction = (new_direction[0],-new_direction[1])

    coords = (coords[0] + new_direction[0], coords[1] + new_direction[1])
    mask = draw_circle(new_circle, R, coord=coords)
    new_circle[mask] = 1
    images.append(new_circle)
    trajectory.append(new_direction)
return images, trajectory

```

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```

1 def detect_directions(prev_img, curr_img):
2     diff = curr_img - prev_img
3     x_coor = np.mean(np.where(diff>0)[0] - np.where(diff<0)[0])
4     y_coor = np.mean(np.where(diff>0)[1] - np.where(diff<0)[1])
5     if abs(x_coor) > abs(y_coor):
6         #смещение по x
7         if x_coor < 0:
8             return (-1, 0)
9         else:
10            return (1, 0)
11    else:
12        #смещение по y
13        if y_coor < 0:
14            return(0, -1)
15        else:
16            return (0, 1)
17    raise Exception("Something wrong!")
18    pred_traj = []
19    for i in range(len(circles)-1):
20        pred_dir = detect_directions(circles[i],circles[i+1])
21        pred_traj.append(pred_dir)
22
23    if (np.array(traj) != np.array(pred_traj)).sum() == 0:
24        print("Верная траектория")
25    else:
26        print("Пока неправильно, попробуйте еще раз")

```

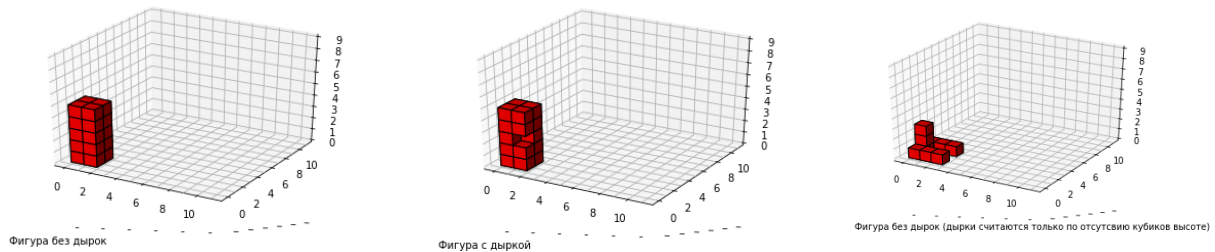
### ***Задача II.3.7. Поиск дырок в трехмерном объекте (26 баллов)***

*Темы: компьютерное зрение.*

#### ***Условие***

Кубитоклобусы прячут найденные в фиджитал-мире ценные вещи в дырки, которые находят в оцифрованных объектах.

Имеется скан некоторого трехмерного объекта. Объект задан в трехмерном пространстве  $(9, 11, 11)$ , где 1 соответствует детали объекта. Вам необходимо найти дырку в объекте, в ответе указать координаты дырки. Дырками объекта считаются пропуски деталей по координате  $z$ . Объясним проще: фигура строится из вертикальных столбиков кубиков. Если в столбике есть промежуток, через который кубики в столбике появляются снова (кубики пропущены в вертикальном ряду), то эта — дырка.



На вход вам будут даны  $N$  отсканированных объектов. Необходимо обнаружить есть ли в них дырки. Если дырок не обнаружено, то вывести ноль, если обнаружены, то вывести их координаты в виде массива. Ответы необходимо вписать в порядке обнаружения.

Пример ответа и фигур. Красным обозначены фигуры, которые даны на входе, голубым изображены дырки в фигурах.

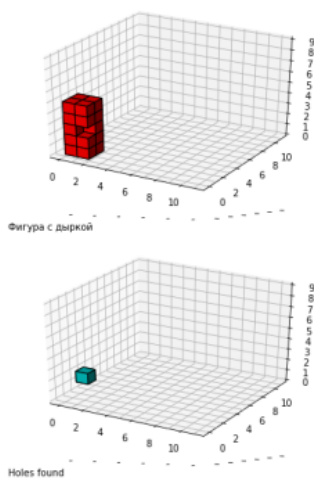


Рис. П.3.3. а)

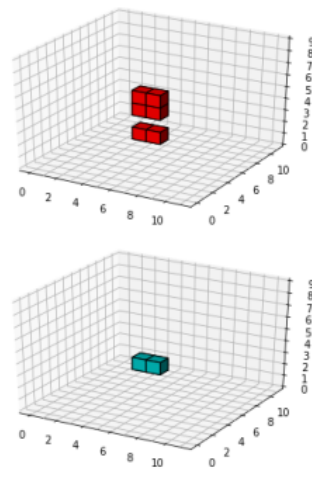


Рис. П.3.4. б)

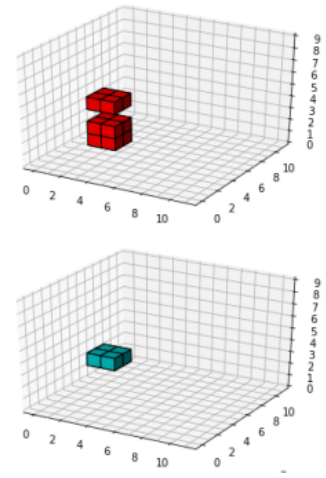


Рис. П.3.5. в)

*#генерируем N простых фигур, и N таких же фигур с дырками*

```
def generate_shapes(N, dims):
    z, x, y = dims
    shapes = []
    for i in range(N):
        shape = np.zeros(shape=(z, x, y))
        z_up = random.randint(4, z-1)
        z_l = random.randint(0, z_up-4)
        x_up = random.randint(1, x-1)
        x_l = random.randint(0, x_up-1)
        y_up = random.randint(1, y-1)
        y_l = random.randint(0, y_up-1)
        shape[z_l:z_up, x_l:x_up, y_l:y_up] = 1
        shapes.append(shape)
    shape_holes = shape.copy()
```



```

        hole_z_up = random.randint(z_l+2, z_up-1)
        hole_z_l = random.randint(z_l+1, hole_z_up-1)
        hole_x_up = random.randint(x_l+1, x_up)
        hole_x_l = random.randint(x_l, hole_x_up-1)
        hole_y_up = random.randint(y_l+1, y_up)
        hole_y_l = random.randint(y_l, hole_y_up-1)
        shape_holes[hole_z_l:hole_z_up, hole_x_l:hole_x_up+1, hole_y_l:hole_y_up+1] = 0
        shapes.append(shape_holes)

    return shapes
def main():
    z, x, y = (9, 11, 11)
    shapes = generate_shapes(5,(z, x, y))
    return shapes

```

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1  # Получаем залитую фигуру(фигуру у которой нет дырок)
2  def get_filled_shape(shape):
3      lower_ends = np.argmax(shape,axis=0)
4      upper_ends = (shape.shape[0] - np.argmax(np.flip(shape,axis=0),axis=0)) %
   ↪ shape.shape[0]
5      filled_shape = np.zeros(shape.shape)
6      for x,y in zip(np.where(upper_ends)[0],np.where(upper_ends)[1]):
7          upper_b = upper_ends[x,y]
8          lower_b = lower_ends[x,y]
9          filled_shape[lower_b:upper_b,x,y] = 1
10     return filled_shape
11 # проверяем есть ли дырки, сравнивая с залитой фигурой
12 def check_holes(shape):
13     filled_shape = get_filled_shape(shape)
14     holes = filled_shape != shape
15     if (holes.sum(>0)):
16         print("Holes found")
17         print(np.argwhere(holes))
18     else:
19         print("No holes found")
20         print("0")
21 for s in shapes:
22     check_holes(s)

```

**Ответ:** а)  $[[2\ 1\ 0]]$ ; б)  $[[4\ 6\ 2], [4\ 7\ 2]]$  в)  $[[4\ 3\ 1], [4\ 3\ 2], [4\ 4\ 1], [4\ 4\ 2]]$ .