

# Интеллектуальные энергетические системы

2022/23 учебный год

## Заключительный этап

### Предметный тур

#### Информатика. 8–11 класс

##### *Задача VI.1.1.1. Производство (10 баллов)*

Для обучения искусственного интеллекта (ИИ) требуются На завод «А» поступило  $t$ -запросов об изготовлении партий рабочих Роботов в количестве  $n$ -штук. Известно, что завод может на каждом этапе производства изготавливать только одного Робота целиком. Каждый этап производства длится  $m$ -часов, поэтому завод начинает производить Роботов непрерывно, пока не изготовит всю партию целиком. Однако по мере освобождения рабочих завода, они запускают процесс изготовления нового Робота. Разница между запусками в производство двух Роботов равна  $p$ -часов. Всего этапов производства на заводе  $k$ .

После того, как Робот сходит с производственного конвейера, его перевозят на склад. При этом Робот успевает оценить, сколько еще других Роботов находится в процессе производства на данный момент, и записывает это число в Базу данных его партии.

Заказчикам интересна максимальная сумма чисел в Базе данных по каждой партии.

##### *Формат входных данных*

В первой строке вводится одно число  $t$  ( $1 \leq t \leq 10^4$ ) — количество запросов, поступивших на завод.

Далее идут  $t$  строк. Каждая строка представляет отдельный запрос в виде последовательности четырех чисел:  $n, m, k, p$  ( $1 \leq n, m, k, p \leq 10^9$ ) — количество Роботов в партии, длительность одного этапа производства, количество этапов производства, интервал между запусками двух Роботов.

##### *Формат выходных данных*

Для каждого запроса необходимо вывести максимальную сумму чисел в Базе данных по партии, а также время, затрачиваемое на изготовление всей партии.

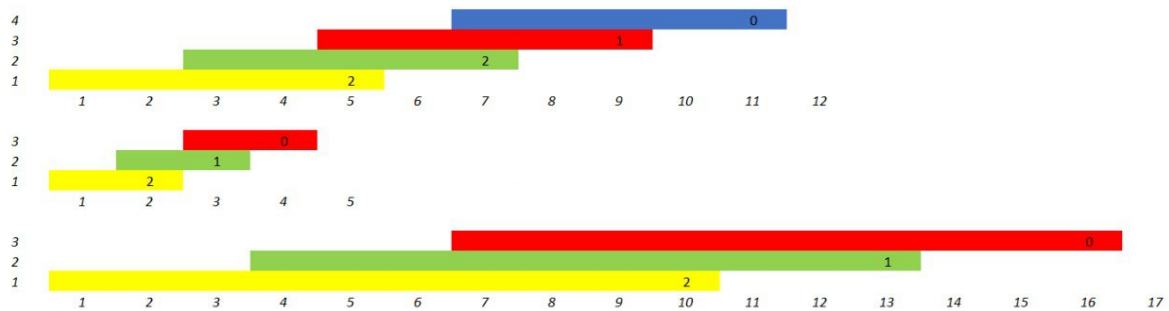
## Примеры

### Пример №1

Стандартный ввод
3
4 5 1 2
3 2 1 1
3 5 2 3

Стандартный вывод
5 11
3 4
3 16



Цвет обозначает в какое время Робот находился на конвейере. Значения в ячейках — число, которое Робот записывает в Базу данных его партии.

## Решение

Заметим, что время производства одного робота равно  $(m \cdot k)$ . Найдем количество Роботов, которых  $i$ -робот посчитает и занесет в Базу данных партии. Это Роботы с номерами  $[i + 1, i + \min(m \cdot k/p, n)]$ . Тогда каждый из первых  $\max(n - m \cdot k/p, 0)$  Роботов запишет число  $(m \cdot k/p)$  в Базу данных, а каждый последующий на 1 меньше, чем предыдущий. Таким образом, ответом на первый вопрос будет:

$$\max(0, n - (m \cdot k)/p) \cdot (m \cdot k)/p + \min(n - 1, (m \cdot k)/p - 1) \cdot \min(n, (m \cdot k)/p)/2.$$

А ответом на второй вопрос будет время, затраченное на изготовление последнего Робота и время до начала его изготовления, т. е.  $((m \cdot k) + p \cdot (n - 1))$ .

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 t = int(input())
2 for i in range(t):
3     n, m, k, p = map(int, input().split())
4     f = min((m * k) // p, n - 1)
5     print(f * (f - 1) // 2 + f * (n - f), p * (n - 1) + m * k)
```



---

### *Формат входных данных*

Первая строка содержит три целых числа  $n$ ,  $X$ ,  $Y$  — число разрушенных блоков с энергетическими элементами, высоту левого и правого края отсека. Вторая строка содержит  $n$  чисел  $H_i$  — высоты столбцов блоков ( $1 \leq n \leq 1000, 1 \leq X, Y, H_i \leq 10^6$ ).

### *Формат выходных данных*

Выведите минимальное число блоков с энергетическими элементами, которое нужно добавить, чтобы у Робота получилось добраться от левого края отсека к правому. Если построить переход нет возможности, выведите число  $-100$ .

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
2 2 2 2 2
<b>Стандартный вывод</b>
0

#### *Пример №2*

<b>Стандартный ввод</b>
3 1 2 3 2 1
<b>Стандартный вывод</b>
-100

### *Решение*

Разберем возможную последовательность действий алгоритма:

1. необходимо найти наибольшие по высоте столбцы блоков;
2. проверить или сделать возможным, чтобы с них можно было сойти влево и вправо;
3. найти следующие наибольшие по высоте блоки, и повторять до тех пор, пока не дойдем до самых низких столбцов блоков;
4. проверить сможет ли Робот перейти отсек слева направо.

У границ отсека смотреть левые и правые спуски соответственно нет необходимости. На оба граничных спуска нельзя накладывать блоки.

### *Пример программы-решения*

Ниже представлено решение на языке Python 3.

```

1 def checker(blocks_list, pr_mx):
2     nw_mx = 0
3     for i in range(n + 2):
4         if blocks_list[i] > nw_mx and blocks_list[i] < pr_mx:
5             nw_mx = blocks_list[i]
6     blocks_index = []
7     for i in range(n + 2):
8         if blocks_list[i] == nw_mx:
9             blocks_index.append(i)
10    return blocks_index, nw_mx
11
12 n, a, b = [int(i) for i in input().split()]
13 blocks = [a] + [int(i) for i in input().split()] + [b]
14 new_blocks = 0
15 maximum = max(blocks) + 1
16 while maximum != min(blocks):
17     ind, maximum = checker(blocks, maximum)
18     for i in ind:
19         if i != 0 and i - 1 != 0 and blocks[i] - blocks[i - 1] > 1:
20             razn = blocks[i] - blocks[i - 1] - 1
21             blocks[i - 1] += razn
22             new_blocks += razn
23         if i != len(blocks) - 1 and i + 1 != len(blocks) - 1 and blocks[i] -
24         ↪ blocks[i + 1] > 1:
25             razn = blocks[i] - blocks[i + 1] - 1
26             blocks[i + 1] += razn
27             new_blocks += razn
28     for i in range(1, n + 2):
29         if abs(blocks[i] - blocks[i - 1]) > 1:
30             print(-100)
31             break
32     else:
33         print(new_blocks)

```

### Задача VI.1.1.3. Спираль (20 баллов)

В блоке «С» имеется бесконечная клетчатая плоскость, в каждую клетку которой записано натуральное число, начиная с 1 по спирали, в соответствии с таблицей. Спираль заполняется по часовой стрелке.

73	74	75	76	77	78	79	80	81
72	43	44	45	46	47	48	49	50
71	42	21	22	23	24	25	26	51
70	41	20	7	8	9	10	27	52
69	40	19	6	1	2	11	28	53
68	39	18	5	4	3	12	29	54
67	38	17	16	15	14	13	30	55
66	37	36	35	34	33	32	31	56
65	64	63	62	61	60	59	58	57

Робот может двигаться вперед или поворачивать на 90 градусов против часовой стрелки. Каждый ход Робота — это перемещение на соседнюю клетку по направлению его движения.

Робот начал движение вниз из клетки с номером 1 бесконечной клетчатой плоскости.

---

Первая итерация движения: Робот идет вперед на 3 клетки, поворачивает, идет вперед 2 клетки, поворачивает, идет вперед 2 клетки, поворачивает, идет вперед 4 клетки, поворачивает (3, 2, 2, 4).

Вторая итерация движения: Робот идет вперед на 8 клеток, поворачивает, идет вперед 6 клеток, поворачивает, идет вперед 4 клетки, поворачивает, идет вперед 8 клеток, поворачивает (8, 6, 4, 8).

Для следующих итераций движения схема повторяется по формуле  $(3 + 5k, 2 + 4k, 2 + 2k, 4 + 4k)$ , где  $k$  — коэффициент количества повторов начального движения.

### *Формат входных данных*

В первой строке вводится число  $n$  ( $10 \leq n \leq 10^{12}$ ).

### *Формат выходных данных*

В первой строке выведите сколько существует точек на спирали, в которых Робот был два раза.

Во второй строке выведите значение клетки на спирали, в которой Робот последний раз бывал неоднократно, после  $n$  ходов.

Примечание: ответ необходимо выводить по модулю  $10^9 + 7$ .

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
10
<b>Стандартный вывод</b>
1
4

#### *Пример №2*

<b>Стандартный ввод</b>
25
<b>Стандартный вывод</b>
1
4

### *Решение*

Заметим, что Робот не бывает более 2 раз в одной клетке спирали. И с каждым увеличением коэффициента  $k$  он спускается все ниже по спирали.

Назовем выделенным красным цветом ячейки «вершиной», а желтым «петлей». Из вершин у нас начинается новая итерация, а в петлях мы получаем неоднократное прохождение Роботом ячейки.

				1				
		12	11	2	9	8		
		13		3		7		
		14		4	5	6		
		15						
38	37	16	35	34	33	32	31	30
39		17						29
40		18						28
41		19						27
42		20	21	22	23	24	25	26
43								

Чтобы прийти из первой вершины в первую петлю, Роботу необходимо сделать следующие ходы: 3 вниз, 2 вправо, 2 вверх и 2 влево, т. е. сделать 9 ходов. Чтобы прийти из второй вершины во вторую петлю, Роботу необходимо сделать следующие ходы: 8 вниз, 6 вправо, 4 вверх и 6 влево, т. е. сделать 24 хода. Чтобы прийти из  $k$  вершины в  $k$  петлю, Роботу необходимо сделать следующие ходы:  $(3 + 5k)$  вниз,  $(2 + 4k)$  вправо,  $(2 + 2k)$  вверх и  $(2 + 4k)$  влево, т. е. сделать  $(9 + 15k)$  ходов.

Заметим, что расстояние между  $k$  петлей и  $(k + 1)$  вершиной ровно два хода. Следовательно, чтобы попасть из  $k$  петли в  $(k + 1)$  петлю, необходимо сделать  $(2 + (9 + 15k)) = (11 + 15k)$  ходов. Таким образом петли появляются на 10, 36, 77 и т. д. ходах. Теперь, зная сколько ходов сделал Робот, можно определить сколько петель он посетил, например, с помощью бинарного поиска, формулы или простого цикла. При  $n = 1\,000\,000\,000$  будет посещено всего 11546 петель.

Заметим, что расстояние по направлению вниз от  $k$  вершины до  $k$  петли равно  $(1 + 3k)$ , т. е. расстояние между  $k$  петлей и  $(k + 1)$  петлей равно  $(1 + 3k)$ . Значит, расстояние между первой петлей и  $k$  петлей равно

$$i = 1k(1 + 3(i - 1)) = k + 3 \cdot (k - 1) \cdot ((k - 1) + 1)/2.$$

Заметим, что расстояние между  $k$  вершиной и  $k$  петлей по направлению влево равно 0, а расстояние между  $k$  петлей и  $(k + 1)$  вершиной равно 2, значит расстояние между  $k$  петлей в  $(k + 1)$  петлей равно 2, т. е. расстояние между первой петлей и  $k$  петлей равно  $2(k - 1)$ .

Введем понятие « $i$  рамка» это  $i$  ободок на спирали. Первый ободок состоит только из 1, второй из  $\{2, 3, 4, 5, 6, 7, 8, 9\}$  и т. д. Для полноты понятия можно рассмотреть рисунок (рамки чередуют свой цвет).

73	74	75	76	77	78	79	80	81
72	43	44	45	46	47	48	49	50
71	42	21	22	23	24	25	26	51
70	41	20	7	8	9	10	27	52
69	40	19	6	1	2	11	28	53
68	39	18	5	4	3	12	29	54
67	38	17	16	15	14	13	30	55
66	37	36	35	34	33	32	31	56
65	64	63	62	61	60	59	58	57

Заметим, что если мы сдвинемся на расстояние по направлению вниз между первой вершиной и последней петлей, то мы попадем в ту же рамку, в которой находится

петля, так как сдвиг по направлению влево растет медленней чем расширяется рамка спирали.

Назовем «началом» рамки клетки, отмеченные зеленым цветом, т. е. те, от которых начинает свое движение рамка, а оранжевым цветом «серединой» рамки, т. е. те, которые находятся строго под началом первой рамки. Также назовем «концом» рамки клетки, отмеченные голубым цветом, т. е. те, в которых заканчивается рамка, причем все значения в этих ячейках являются полными квадратами нечетных чисел. При этом отметим, что 1 является началом и концом рамки.

73	74	75	76	77	78	79	80	81
72	43	44	45	46	47	48	49	50
71	42	21	22	23	24	25	26	51
70	41	20	7	8	9	10	27	52
69	40	19	6	1	2	11	28	53
68	39	18	5	4	3	12	29	54
67	38	17	16	15	14	13	30	55
66	37	36	35	34	33	32	31	56
65	64	63	62	61	60	59	58	57

Теперь отметим, что разность значений в начале и середине  $(i + 1)$  рамки на 3 больше, чем в  $i$  рамке, т. е. оно растет с каждым разом на 3. А разность значений между началом  $(i + 1)$  рамки и концом  $i$  рамки равно 1. Таким образом, разности значений между концом и серединой первых соседних рамок равны: 3, 6, 9, 12, ...

Зная, в какой рамке находится последняя петля (расстояние между первой вершиной и последней петлей по направлению вниз), какое количество ходов было совершено от середины рамки (расстояние между первой вершиной и последней петлей по направлению влево), а также разность значений между концом предыдущей рамки и серединой текущей, можно легко вычислить значение клетки спирали, в которой была последняя петля — это будет сумма (значения конца предыдущей рамки) + (разность значений между концом предыдущей рамки и серединой текущей) + (количество ходов было совершено от середины рамки по направлению влево).

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 t = 10
3 k = 0
4 while t <= n:
5     k += 1
6     t += 11 + 15 * k
7 print(k)
8 y = k + 3 * ((k - 1) * ((k - 1) + 1)) // 2
9 x = 2 * (k - 1)
10 mod = 10 ** 9 + 7
11 ram = (2 * y - 1) % mod
12 res = (ram * ram + 3 * y + x) % mod
13 print(res)
```



### Задача VI.1.1.4. Распределение энергии (25 баллов)

На станции «D», сдерживающей перебои в системе электроснабжения, необходимо поставить несколько систем распределения электрической энергии. Причем каждая следующая система распределения должна быть не менее, чем в  $k$  раз мощнее предыдущей, чтобы избежать быстрого износа станции. Для этого туда был отправлен Робот, он может построить данные системы распределения. Руководству станции стало интересно, сколько вариантов построения станции существует, если суммарная мощность систем, которые может построить Робот, будет  $n$ .

#### Формат входных данных

Два целых числа  $n$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $2 \leq k \leq n$ ) — суммарная мощность систем, коэффициент мощности между соседними станциями.

#### Формат выходных данных

Выведите одно число — сколько вариантов построения станции существует.

Примечание: мощность одной системы распределения электрической энергии выражается целым числом. Суммарная мощность систем должна быть  $n$ .

Примечание: ответ необходимо выводить по модулю  $10^9 + 7$ .

#### Примеры

##### Пример №1

Стандартный ввод
2 2
Стандартный вывод
1

##### Пример №2

Стандартный ввод
5 2
Стандартный вывод
2

#### Решение

Данная задача является задачей динамического программирования. Решение использует двумерное ДП  $q[a][b]$ , хранящее количество способов построения станции с  $b$  системами суммарной мощностью  $a$ . В задаче, можно рассчитать ДП сразу для всех тестов при фиксированном  $k$ , а затем выдавать ответ.

Базой ДП будет, что станция, состоящая из 1 системы при любой мощности равна всего 1. Переходом будет сумма всех способов для станций с длиной не большей,

---

чем рассматриваемая и мощностью не больше, чем рассматриваемая, без учета минимальной мощности, необходимой для станции данной длины (сумма  $k^i$  от  $i = 0$  до рассматриваемая длина). Ответом на вопрос будет сумма всех переходов ДП при мощности  $n$ .

Докажем переход. Минимальная станция, состоящая из  $i$ -систем выглядит так:  $\{1, k, k^2, \dots, k^{i-1}\}$ , теперь отметим, что если мы возьмем станции состоящие из не более, чем  $i$  станций, например  $\{c_0, c_1, c_2, \dots, c_{i-1}\}$  и сложим соответствующие элементы, то получим станцию, удовлетворяющую условию  $\{1 + c_0, k + c_1, k^2 + c_2, \dots, k^{i-1} + c_{i-1}\}$ .

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 import math
2
3 n, k = map(int, input().split())
4 x, y = int(math.log(n, k)) + 2, 100005
5 q = [[0 for i in range(x)] for o in range(y)]
6 w = [k ** i for i in range(x)]
7 mod = 10**9 + 7
8 for i in range(y):
9     q[i][1] = 1
10 for o in range(2, x):
11     t = sum(w[:o])
12     for i in range(t, y):
13         q[i][o] = sum(q[i - t][:o + 1]) % mod
14 print(sum(q[n]) % mod)
```

### Задача VI.1.1.5. Разделение системы (25 баллов)

На станции «Е» на работу одной системы расходуется  $a_i^2$  электрической энергии, где  $a_i$  — это размер  $i$ -системы, причем  $a_i$  — целое число. Изначально на станции было  $n$ -систем. Станция максимально может обслуживать  $k$ -систем одновременно. Начальник станции хочет, чтобы суммарная расходуемая электрическая энергия на работу всех систем была минимальна. Помогите Роботу, перестроить систему так, чтобы соблюдались требования руководства.

Примечание: под перестройкой системы подразумевается разделение ее на системы меньшего размера, суммарно сохраняющие размер системы.

#### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $k$  ( $1 \leq n \leq k \leq 10^5$ ) — изначально количество систем на станции и максимальное количество систем, которое может обслужить станция.

В следующей строке находятся  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6$ ) — размеры систем.

Известно, что системы на станции можно распределить так, чтобы станция обслуживала все  $k$ -систем одновременно.

## Формат выходных данных

Выведите одно целое число — минимальную расходуемую электрическую энергию на работу всех систем.

## Примеры

Один цвет — это разбиение системы на меньшие по размеру. Значения в ячейках — это электрическая энергия на работу данной системы.

### Пример №1

<b>Стандартный ввод</b>
3 6
5 3 1
<b>Стандартный вывод</b>
15

	25					9					1
4		4		1		4		1		1	

### Пример №1

<b>Стандартный ввод</b>
1 4
19
<b>Стандартный вывод</b>
91

	361											
	25			25			25					16

## Решение

Заметим, что если количество систем на станции уже достигло значения  $k$ , то мы не можем перестраивать системы.

Также стоит отметить, что перестраивать системы стоит на системы примерно одинакового размера, например если станцию размера 15 хотим перестроить в 3 станции, то это будут системы размерами  $\{5, 5, 5\}$ , а если в 2 системы, то это будут системы размерами  $\{7, 8\}$ . Поэкспериментировать с данным пунктом можно самостоятельно.

Введем для каждой системы два параметра: размер системы и на какое количество мы будем ее перестраивать ( $len, nums$ ). Теперь для каждой системы мы можем произвести подсчет на сколько нам выгодно перестраивать каждую систему. После чего мы будем выбирать каждый раз, пока итоговое количество систем не станет

---

равно  $k$ , для выбора самой выгодной системы для перестройки можно использовать самые разные алгоритмы, например куча, дерево отрезков и другие. Пересчитываем выгодность системы как разность затрачиваемой энергии станциями, если их  $nums$ , и если их  $nums + 1$ . Для станции размером  $len$  ( $len = a_i$ ) и  $nums$  перестройка будет использовать энергии:

$$\left( nums - \frac{len}{nums} \cdot nums \right) \cdot \frac{len}{nums} \cdot \frac{len}{nums} + \left( \frac{len}{nums} \cdot nums \right) \cdot \left( \frac{len}{nums} + 1 \right) \cdot \left( \frac{len}{nums} + 1 \right),$$

деление целочисленное с округлением вниз.

После того, как систем на станции стало ровно  $k$ , считаем ответ как сумма

$$\left( nums - \frac{len}{nums} \cdot nums \right) \cdot \frac{len}{nums} \cdot \frac{len}{nums} + \left( \frac{len}{nums} \cdot nums \right) \cdot \left( \frac{len}{nums} + 1 \right) \cdot \left( \frac{len}{nums} + 1 \right)$$

для всех систем на станции.

### **Пример программы-решения**

Ниже представлено решение на языке Python 3.

```
1 from heapq import *
2
3 n, k = map(int, input().split(" "))
4
5 def val(l, nums):
6     unit = l // nums
7     extra = l - unit * nums
8     return (nums - extra) * unit * unit + extra * (unit + 1) * (unit + 1)
9
10 pq = []
11 arr = list(map(int, input().split(" ")))
12 total = 0
13 for x in range(n):
14     total += arr[x] * arr[x]
15     heappush(pq, (-val(arr[x], 1) + val(arr[x], 2), arr[x], 2))
16
17 for x in range(k - n):
18     temp = heappop(pq)
19     total += temp[0]
20     a, b = temp[1], temp[2]
21     heappush(pq, (-val(a, b) + val(a, b + 1), a, b + 1))
22
23 print(total)
```

Тестовые наборы для задач представлены по ссылке — [https://disk.yandex.ru/d/RMHm\\_8SH5J4cXQ](https://disk.yandex.ru/d/RMHm_8SH5J4cXQ).

---

## Математика. 8–9 классы

### Задача VI.1.2.1. (20 баллов)

Темы: элементы теории чисел.

Найти все простые числа  $p$  и  $q$  такие, что  $p^3 - q^3 = 3p^2q - 3pq^2 + p + q$ .

#### Решение

Преобразовав исходное равенство по формуле куб разности, получим

$$(p - q)^3 = p + q.$$

Пусть  $n = p - q$ , тогда  $n^3 = n + 2q$ . Отсюда

$$q = \frac{n^3 - n}{2} = \frac{(n - 1)n(n + 1)}{2}.$$

Среди трех последовательных чисел одно делится на 3, поэтому  $q$  делиться на 3. Так как  $q$ -простое, то  $q = 3$ , это значение получается при  $n = 2$ .

Тогда  $p = 5$ .

Ответ:  $q = 3$ ,  $p = 5$ .

#### Критерии оценивания

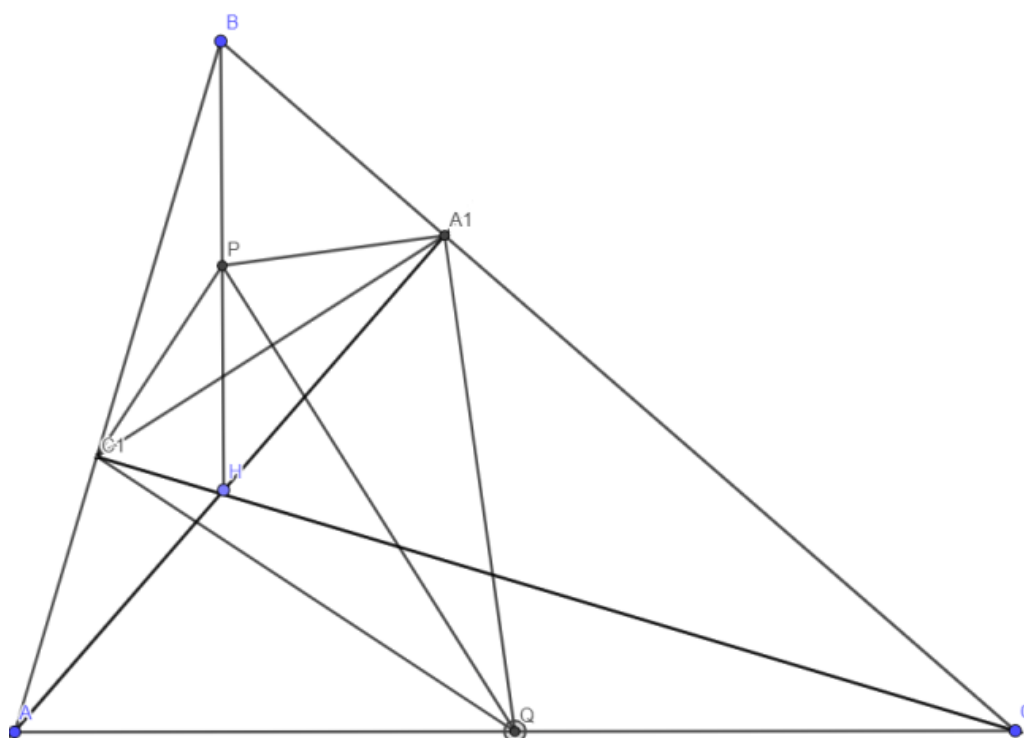
- Простые числа  $p$  и  $q$  найдены подбором — 0 баллов.
- В исходном выражении выделен полный куб разности — 5 баллов.
- Сделан вывод о том, что  $q$  кратно 3 — 5 баллов.
- Сделан вывод, что  $q = 3$  — 6 баллов.
- Найдено число  $p = 5$  — 5 баллов.
- Приведено полное обоснованное решение — 20 баллов (как сумма всех предыдущих продвижений).

### Задача VI.1.2.2. (25 баллов)

Темы: геометрия.

В остроугольном треугольнике  $ABC$  точки  $A_1$  и  $C_1$  — основания высот, проведённых из вершин  $A$  и  $C$  соответственно,  $H$  — точка пересечения высот,  $P$  и  $Q$  — середины отрезков  $BH$  и  $AC$ . Найти угол между прямыми  $PQ$  и  $C_1A_1$ .

## Решение



В прямоугольных треугольниках  $AC_1C$  и  $AA_1C$  отрезки  $C_1Q$  и  $A_1Q$  являются медианами к гипотенузе  $AC$ . Следовательно,  $C_1Q = A_1Q = \frac{1}{2}AC$  (по свойству медианы прямоугольного треугольника, проведенной из вершины прямого угла).

Аналогично в прямоугольных треугольниках  $BC_1H$  и  $BA_1H$  отрезки  $C_1P$  и  $A_1P$  являются медианами к гипотенузе  $BH$ . Следовательно,

$$C_1P = A_1P = \frac{1}{2}BH.$$

Таким образом, треугольники  $A_1QC_1$  и  $A_1PC_1$  — равнобедренные. Следовательно серединный перпендикуляр к отрезку  $A_1C_1$  является в обоих треугольниках медианой из вершин  $P$  и  $Q$  соответственно. Следовательно,  $PQ$  перпендикулярен  $A_1C_1$  и угол между ними равен  $90^\circ$ .

**Ответ:**  $90^\circ$ .

### Критерии оценивания

- Доказано, что треугольник  $A_1QC_1$  равнобедренный — 8 баллов.
- Доказано, что треугольник  $A_1PC_1$  равнобедренный — 8 баллов.
- Сделан вывод о том, что  $PQ$  содержит серединные перпендикуляры в треугольниках  $A_1QC_1$  и  $A_1PC_1$  — 9 вывод.
- Приведено полное обоснованное решение — 25 баллов (как сумма всех предыдущих продвижений).

---

### Задача VI.1.2.3. (25 баллов)

Темы: системы уравнений.

Решить систему уравнений

$$\begin{cases} \sqrt{xy} - 29 = \sqrt{x} + \sqrt{y} \\ x - \sqrt{x} = \sqrt{y} - y + 72 \end{cases}$$

#### Решение

Заметим, что  $x \geq 0$ ,  $y \geq 0$  и преобразуем исходную систему к виду

$$\begin{cases} \sqrt{xy} - 29 = \sqrt{x} + \sqrt{y} \\ x + y = \sqrt{x} + \sqrt{y} + 72 \end{cases}$$

Введя замену

$$\begin{cases} \sqrt{x} + \sqrt{y} = u, \\ \sqrt{xy} = v, \end{cases}$$

$u \geq 0$ ,  $v \geq 0$ , получим, что  $x + y = u^2 - 2v$  и исходная система примет вид

$$\begin{cases} v - 29 = u \\ u^2 - 2v = u + 72. \end{cases}$$

Выразив из первого уравнения  $v = u + 29$ , подставив его во второе уравнение системы и приведя подобные слагаемые, мы получим квадратное уравнение

$$u^2 - 3u - 130 = 0.$$

Корнями данного уравнения являются  $u_1 = -10$ ,  $u_2 = 13$ . Первый корень является посторонним из-за ограничения на переменные. Тогда получаем, что

$$\begin{cases} u = 13 \\ v = 13 + 29 = 42 \end{cases}$$

Возвращаемся обратно к замене

$$\begin{cases} \sqrt{x} + \sqrt{y} = 13, \\ \sqrt{xy} = 42 \end{cases}$$

Решая данную систему, получим два набора  $\sqrt{x} = 6$ ,  $\sqrt{y} = 7$  и  $\sqrt{x} = 7$ ,  $\sqrt{y} = 6$ . Откуда получаем, два решения исходной системы  $x_1 = 36$ ,  $y_1 = 49$  и  $x_2 = 49$ ,  $y_2 = 36$ .

**Ответ:**  $x_1 = 36$ ,  $y_1 = 49$  и  $x_2 = 49$ ,  $y_2 = 36$ .

#### Критерии оценивания

- Решение системы найдено подбором — 0 баллов.

- Введена замена, приводящая исходное уравнение к более простому виду, дальнейших продвижений нет — 5 баллов.
- Новая система решена относительно новых переменных — но не совершен переход к переменным  $x, y$  — 10 баллов.
- Сделан обратный переход к переменным  $x, y$  и получено верное решение — 10 баллов.
- Приведено полное обоснованное решение — 25 баллов (как сумма всех предыдущих продвижений или аналогичное ему).

### Задача VI.1.2.4. (30 баллов)

Темы: комбинаторика.

В городской Думе 20 депутатов. Им предстоит выбрать председателя, спикера, вице-спикера, а также сформировать 5 комитетов, состоящих из не менее 3 депутатов. Каждый депутат должен занять одну и только одну должность. Сколькими способами это можно сделать?

Примечание: если в ходе вычислений получаются факториалы, то нет необходимости в их вычислении.

#### Решение

Председателя, спикера и вице-спикера можно выбрать  $20 \cdot 19 \cdot 18 = 6840$  способами.

Распределим оставшихся депутатов. Возможны два варианта. Первый, когда в одном комитете 5, а в остальных по 3. Пусть 5 человек в первом комитете, тогда количество способов будет

$$C_{17}^5 C_{12}^3 C_9^3 C_6^3 C_3^3 = \frac{17!}{5!12!} \frac{12!}{3!9!} \frac{9!}{3!6!} \frac{6!}{3!3!} = \frac{17!}{5!(3!)^4}.$$

Аналогично для других 4 комитетов, т. е. для первого варианта имеем

$$\frac{5 \cdot 17!}{5!(3!)^4} = \frac{17!}{4!(3!)^4}.$$

Второй вариант, когда в двух комитетах по 4, а в трех по 3. Рассуждая аналогично, получим

$$C_5^2 \cdot C_{17}^4 C_{13}^4 C_9^3 C_6^3 C_3^3 = 10 \cdot \frac{17!}{4!13!} \frac{13!}{4!9!} \frac{9!}{3!6!} \frac{6!}{3!3!} = 10 \cdot \frac{17!}{(4!)^2(3!)^3}.$$

Суммарно имеем

$$6840 \cdot \left( \frac{17!}{(4!(3!)^4} + 10 \cdot \frac{17!}{(4!)^2(3!)^3} \right) = \frac{665 \cdot 17!}{(3!)^3}.$$

Ответ:  $\frac{665 \cdot 17!}{(3!)^3}$ .



---

### Критерии оценивания

- Подсчитано количество способов выбора председателя, спикера и вице-спикера — 5 баллов.
- Рассмотрена ситуация, когда в одном комитете 5 человек, а оставшихся по 3 человека — 10 баллов.
- Рассмотрена ситуация, когда в двух комитетах по 4 человека, а в трех по 3 — 10 баллов.
- Найдено общее количество способов выбора — 5 баллов.
- Приведено полное обоснованное решение — 30 баллов (как сумма всех предыдущих продвижений).

## Математика. 10–11 классы

### Задача VI.1.3.1. (20 баллов)

Темы: элементы теории чисел.

Сумма двух действительных чисел равна 3. Может ли сумма их квадратов равняться 4,4?

#### Решение

Обозначим первое число за  $x$ , тогда второе будет равно  $y = 3 - x$ , а их произведение  $x(3 - x) = -x^2 + 3x$ . Максимальное значение данного выражения достигается при  $x = 1,5$  и составляет 2,25. Тогда

$$x^2 + y^2 = (x + y)^2 - 2xy \geq 9 - 4,5, \quad x^2 + y^2 \geq 4,5.$$

Ответ: нет.

### Критерии оценивания

- Получена формула суммы квадратов двух исходных чисел — 5 баллов.
- Получено выражение для произведения двух исходных чисел — 5 баллов
- Найдено максимальное значение произведения двух исходных чисел — 5 баллов.
- Построена нижняя оценка для суммы квадратов двух исходных чисел — 5 баллов.
- Приведено полное обоснованное решение — 20 баллов (как сумма всех предыдущих продвижений).

### Задача VI.1.3.2. (20 баллов)

Темы: тождественные преобразования.

Вычислить сумму

$$\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \frac{1}{7 \cdot 10} + \dots + \frac{1}{2023 \cdot 2026}.$$

---

**Решение**

Рассмотрим каждое слагаемое по отдельности

$$\begin{aligned}\frac{1}{1 \cdot 4} &= \frac{1}{3} \cdot \frac{3}{1 \cdot 4} = \frac{1}{3} \cdot \left(1 - \frac{1}{4}\right) \\ \frac{1}{4 \cdot 7} &= \frac{1}{3} \cdot \frac{3}{4 \cdot 7} = \frac{1}{3} \cdot \left(\frac{1}{4} - \frac{1}{7}\right) \\ &\dots\dots\dots \\ \frac{1}{2023 \cdot 2026} &= \frac{1}{3} \cdot \frac{3}{2023 \cdot 2026} = \frac{1}{3} \cdot \left(\frac{1}{2023} - \frac{1}{2026}\right)\end{aligned}$$

Тогда

$$\begin{aligned}&\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \frac{1}{7 \cdot 10} + \dots + \frac{1}{2023 \cdot 2026} = \\ &= \frac{1}{3} \cdot \left(1 - \frac{1}{4}\right) + \frac{1}{3} \cdot \left(\frac{1}{4} - \frac{1}{7}\right) + \dots + \frac{1}{3} \cdot \left(\frac{1}{2023} - \frac{1}{2026}\right) = \\ &= \frac{1}{3} \cdot \left(1 - \frac{1}{4} + \frac{1}{4} - \frac{1}{7} + \frac{1}{7} - \frac{1}{10} + \dots + \frac{1}{2023} - \frac{1}{2026}\right) = \\ &= \frac{1}{3} \cdot \left(1 - \frac{1}{2026}\right) = \frac{1}{3} \cdot \frac{2025}{2026} = \frac{675}{2026}.\end{aligned}$$

Ответ:  $\frac{675}{2026}$ .

**Критерии оценивания**

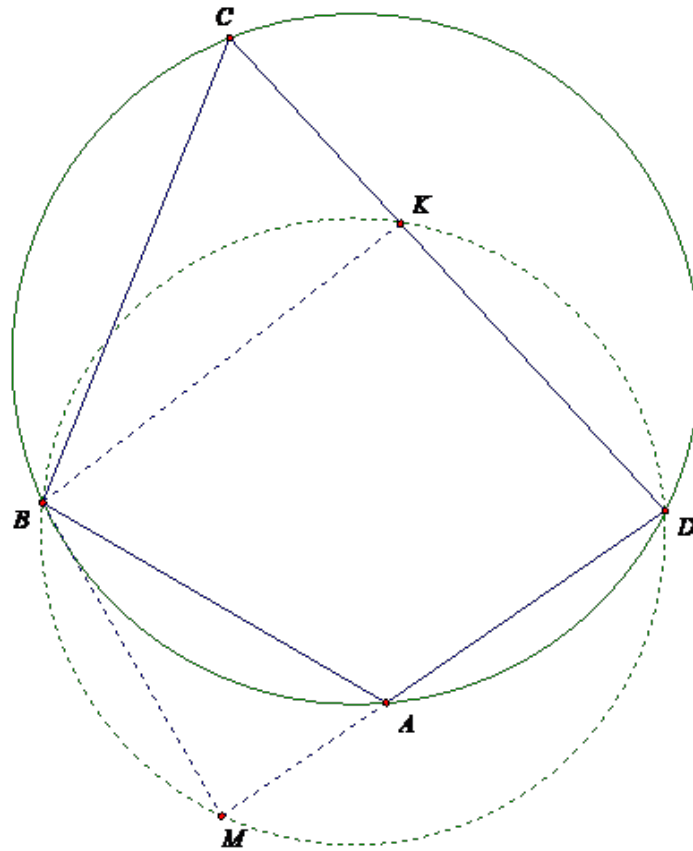
- Каждое слагаемое исходной суммы представлено в виде разности двух дробей — 7 баллов.
- При верной последовательности действий исходная сумма подсчитана, но допущена арифметическая ошибка — штраф 2 балла.
- Приведено полное обоснованное решение — 20 баллов.

**Задача VI.1.3.3. (25 баллов)**

Темы: планиметрия.

На стороне  $CD$  вписанного выпуклого четырехугольника  $ABCD$  отмечена точка  $K$  так, что  $CK = t$ . Около треугольника  $BKD$  описана окружность, пересекающая прямую  $DA$  в точке  $M$  не совпадающей с точкой  $D$ . Найдите длину отрезка  $AM$ , если  $AB = a$ ,  $BC = b$ .

## Решение



Так как  $ABCD$  — вписанный выпуклый четырехугольник, то  $\angle BCD + \angle DAB = 180^\circ$ .

В силу смежности углов:  $\angle DAB + \angle MAB = 180^\circ$ .

Следовательно,  $\angle BCD = \angle MAB$ .

Далее, в силу вписанности четырехугольника  $ABCD$  имеем, что  $\angle CBA + \angle CDA = 180^\circ$ .

Кроме этого, четырехугольник  $KBMD$  также вписанный и  $\angle KBM + \angle KDM = \angle KBM + \angle CDA = 180^\circ$ . Следовательно,  $\angle CBA = \angle KBM$ . Следовательно,  $\angle CBK = \angle ABM$  (так как дополняются  $\angle KBA$  до равных углов).

Тогда имеем, что  $\triangle ABM \sim \triangle CBK$  (по двум углам).

Следовательно,  $\frac{CK}{AM} = \frac{CB}{AB}$ . Отсюда получаем:  $AM = \frac{CK \cdot AB}{BC} = \frac{ma}{b}$ .

Ответ:  $AM = \frac{ma}{b}$ .

### Критерии оценивания

- Доказано равенство углов  $\angle BCD = \angle MAB$  — 5 баллов.
- Доказано равенство углов  $\angle CBA = \angle KBM$  — 5 баллов.
- Доказано равенство углов  $\angle CBK = \angle ABM$  — 5 баллов.

- Доказано подобие треугольников  $ABM$  и  $CBK$  — 7 баллов.
- Найдено верное значение длины отрезка  $AM$  — 3 балла.
- Приведено полное обоснованное решение — 25 баллов (как сумма всех предыдущих продвижений).

### Задача VI.1.3.4. (35 баллов)

Темы: тригонометрия.

Изобразите на координатной плоскости множество решений неравенства:

$$\cos 2x(\sin y + \cos 4y) - 1 \geq 1 - \sin 2x(\cos y + \sin 4y).$$

#### Решение

Раскрывая скобки и группируя, получим

$$\sin(2x + y) + \cos(4y - 2x) \geq 2.$$

Так как  $\sin(2x + y) \leq 1$  и  $\cos(4y - 2x) \leq 1$ , то решением неравенства является система:

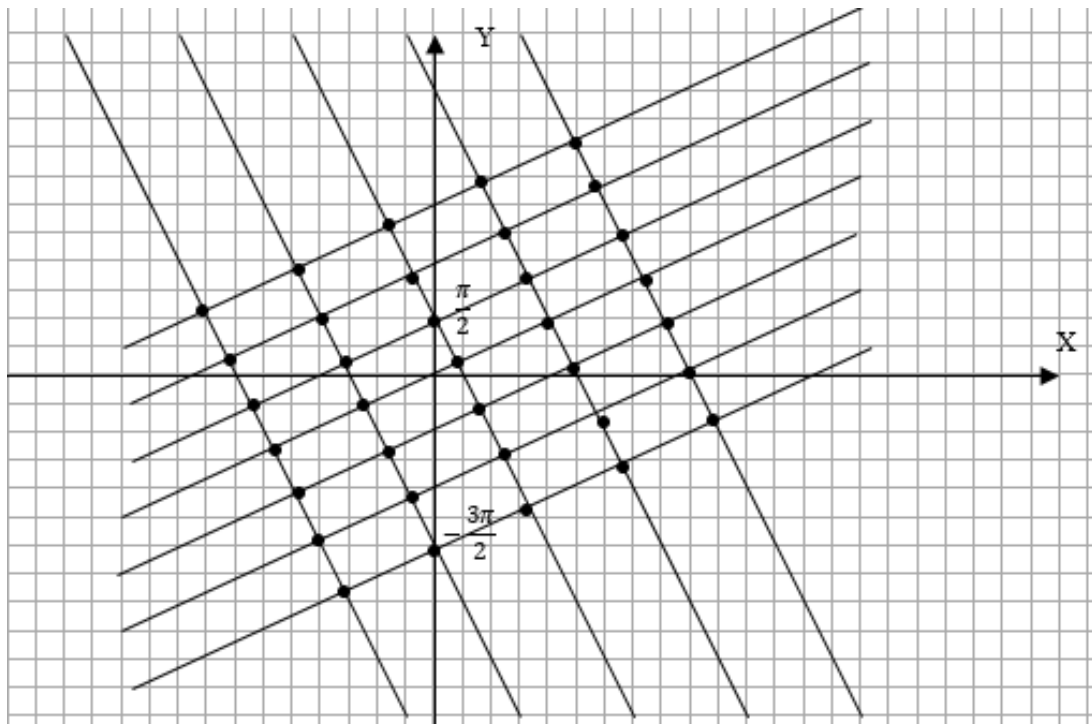
$$\begin{cases} \sin(2x + y) = 1 \\ \cos(4y - 2x) = 1 \end{cases}$$

или

$$\begin{cases} y = -2x + \frac{\pi}{2} + 2k_1\pi \\ y = \frac{x}{2} + \frac{k_2\pi}{2} \end{cases}$$

Решением системы является множество точек плоскости.

Ответ:



---

### *Критерии оценивания*

- Исходное неравенство сведено к виду  $\sin(2x + y) + \cos(4y - 2x) \geq 2$  — 7 баллов.
- Неравенство сведено к решению системы уравнений 
$$\begin{cases} \sin(2x + y) = 1 \\ \cos(4y - 2x) = 1 \end{cases}$$
 — 8 баллов.
- Найдено множество решений исходного неравенства — 8 баллов.
- Построено множество решений исходного неравенства — 12 баллов.
- Приведено полное обоснованное решение — 35 баллов (как сумма всех предыдущих продвижений).