

# Заключительный этап

## Индивидуальный предметный тур

Информатика. 8–11 класс

### *Задача III.1.1.1. (10 баллов)*

Дана матрица состоящая из 0 и 1. Значениями 1 в матрице нарисованы квадраты и круги (рисунок III.1.1). Фигур других форм нет, минимальный размер  $3 \times 3$ . Необходимо определить количество кругов и квадратов. Необходимо определить количество кругов и квадратов.

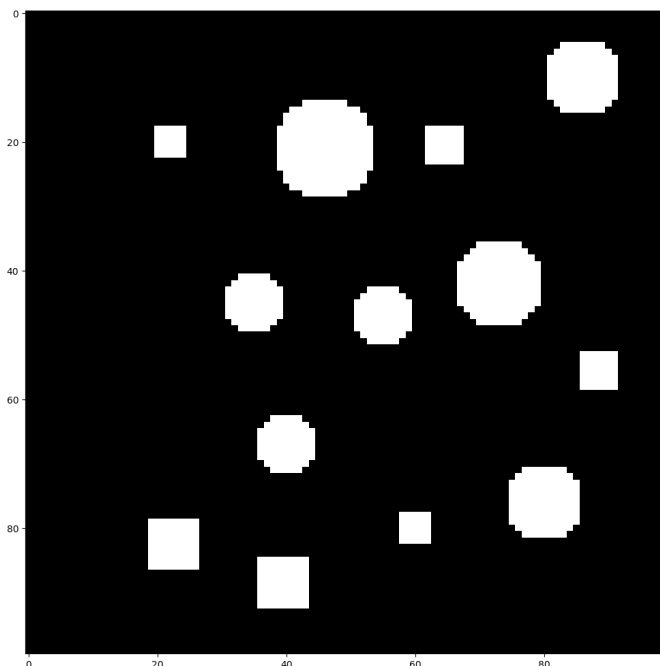


Рис. III.1.1: Пример входного изображения. Белый цвет — фигуры. Черный цвет — пустота

### *Формат входных данных*

В первой строке через пробел заданы высота  $h$  и ширина  $w$  матрицы ( $50 \leq (h, w) \leq 300$ ).

В следующих строках заданы значения матрицы по строкам и столбцам. Фигуры не соприкасаются.

***Формат выходных данных***

Два числа разделенные пробелом: количество квадратов и кругов.

***Примеры***

*Пример №1*



### Пример программы-решения

Ниже представлено решение на языке Python.

```
1 def neighbors(i, j):
2     return ((i - 1, j),
3             (i + 1, j),
4             (i, j - 1),
5             (i, j + 1))
6
7 def _label(image, i, j, label):
8     image[i][j] = label
9     for ni, nj in neighbors(i, j):
10        if image[ni][nj] == -1:
11            _label(image, ni, nj, label)
12
13 def label_image(image):
14     for i in range(len(image)):
15         for j in range(len(image[0])):
16             image[i][j] *= -1
17     label = 0
18     for i in range(len(image)):
19         for j in range(len(image[0])):
20             if image[i][j] == -1:
21                 label += 1
22                 _label(image, i, j, label)
23     return label
24
25 def count_area(labeled, label):
26     area = 0
27     for i in range(len(labeled)):
28         for j in range(len(labeled[0])):
29             if labeled[i][j] == label:
30                 area += 1
31     return area
32
33 def count(image):
34     max_label = label_image(image)
35     n_circles = 0
36     n_squares = 0
37     for i in range(1, max_label+1):
38         area = count_area(image, i)
39         sq = area ** 0.5
40         if sq - int(sq) == 0:
41             n_squares += 1
42         else:
43             n_circles += 1
44     return n_squares, n_circles
45
46 h, w = map(int, input().split())
47 image = []
48
49 for i in range(h):
50     line = map(int, input().split())
51     image.append(list(line))
52
53 ns, nc = count(image)
54
55 print(ns, nc)
```



**Стандартный вывод**

36 37

Пример вывода с сортировкой.

**Стандартный вывод**

```
147 258
168 22
168 269
172 219
183 249
```

**Пример программы-решения**

Ниже представлено решение на языке Python.

```

1 import matplotlib.pyplot as plt
2
3 def neighbors(i, j):
4     return ((i - 1, j),
5             (i + 1, j),
6             (i, j - 1),
7             (i, j + 1))
8
9
10 def find_centers(image):
11     coords = []
12     for row in range(1, len(image) - 1):
13         for col in range(1, len(image[0]) - 1):
14             cval = image[row][col]
15             nvals = []
16             for nrow, ncol in neighbors(row, col):
17                 val = image[nrow][ncol]
18                 nvals.append(val < cval)
19             if sum(nvals) == 4:
20                 coords.append((row, col))
21     return coords
22
23 h, w = map(int, input().split())
24 image = []
25
26 for i in range(h):
27     line = map(float, input().split())
28     image.append(list(line))
29
30 coords = find_centers(image)
31
32 for row, col in sorted(coords):
33     print(row, col)

```

**Задача III.1.1.3. (30 баллов)**

В компьютерной графике и разработке игр, в том числе и для виртуальной реальности, часто приходится решать задачу по движению объектов и определения взаимодействия объектов друг с другом. Самый простой вид взаимодействия это столкновение различных объектов, а самый простой вид движения это линейное движение с заданной скоростью. Необходимо написать программу, которая определит

сколько точек попало в заданные фигуры после полной остановки. Фигуры заданы набором параметров и могут представлять прямоугольник, круг и треугольник. Точки двигаются с заданной скоростью по двум осям ( $x$  и  $y$ ), количество шагов ограничено. Точки перемещаются в ограниченном поле, при столкновении с границей точка меняет свое направление.

На рисунке III.1.3 представлен пример того, как двигаются точки и смена их направления при достижении границ поля. На примере одна точка остановилась на фигуре, вторая остановилась на пустом месте. Если точка остановилась на границе, то это также считается остановкой на фигуре.

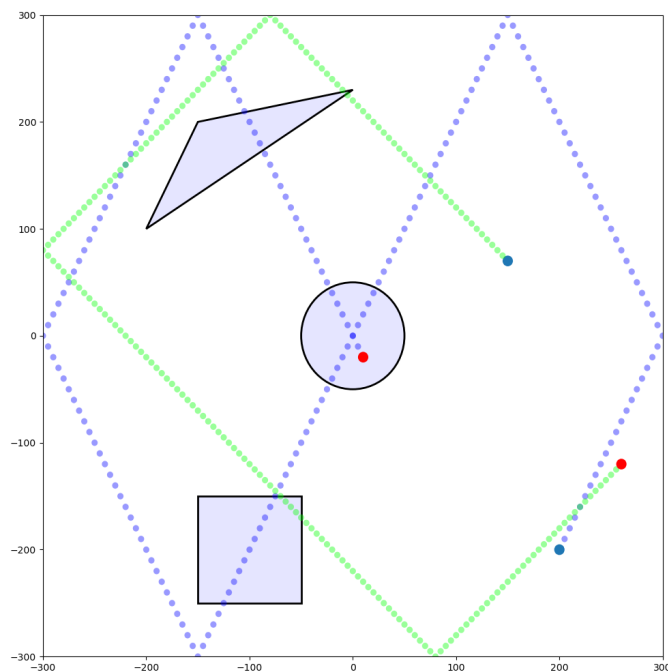


Рис. III.1.3: Пример движения двух точек по полю. Синяя точка — точка начала движения, красная точка — точка окончания движения. Синие и зеленые точки — траектории движения двух разных точек

Фигуры могут выходить за поле и могут пересекаться, тогда считается одно попадание (рисунок III.1.4).

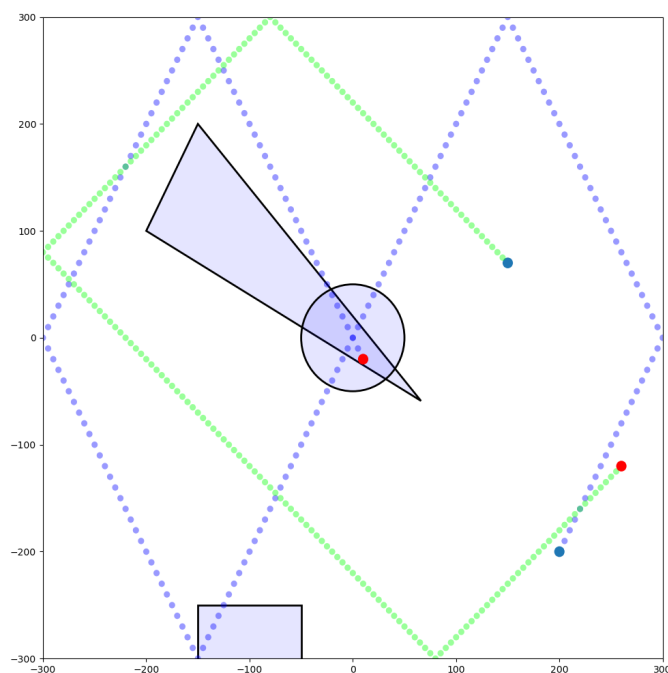


Рис. III.1.4: Пример движения двух точек по полю. Синяя точка — точка начала движения, красная точка — точка окончания движения. Синие и зеленые точки — траектории движения двух разных точек

### **Формат входных данных**

В первой строке через пробел заданы количество размеров поля  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ . Во второй строке через пробел количество фигур  $n_f$  и количество точек  $n_p$ . В следующих  $n_f$  строках идут параметры фигур, каждая фигура на новой строке. Круг задан координатами  $x_c$  и  $y_c$  центра и его радиусом  $r_c$ . Треугольник задан тремя парами точек  $x_1, y_1, x_2, y_2, x_3, y_3$ . Прямоугольник задан координатами  $x_r$  и  $y_r$  левого нижнего угла и размерами по высоте  $h_r$  и ширине  $w_r$ . Далее идет  $n_p$  строк с описанием точек. Каждая точка в новой строке. Точка задается координатами  $x_p$  и  $y_p$  начального положения, скоростью  $v_{x_p}$  и  $v_{y_p}$ , а также количеством шагов  $s_p$ , которая точка должна пройти до остановки.

### **Формат выходных данных**

Количество точек попавших на фигуру в виде одного целого числа.

### **Примеры**

#### *Пример №1*



**Стандартный ввод**

```

-1000 1000 -1000 1000
1 32
0 0 800
593 -145 0.1 -0.25 5450
227 215 1.0 -1.0 7510
-72 -636 0.2 -0.2 3860
-621 -142 -0.2 0.25 5750
-485 618 1.0 -0.2 6360
775 698 -0.2 0.5 970
-853 -229 -0.2 0.25 9610
902 272 -1.0 -0.1 7590
901 -287 -1.0 -1.0 5120
240 -118 0.25 0.25 6950
-922 27 0.25 -0.25 1710
398 -68 0.2 0.1 6400
-800 283 0.25 1.0 8950
836 449 1.0 1.0 830
540 -784 0.5 1.0 1860
-747 936 1.0 -0.25 7240
857 946 -0.25 -0.5 2520
638 614 0.1 -0.2 6780
-79 -408 -0.25 1.0 5990
-302 770 -0.2 0.5 980
-56 -392 0.5 0.1 3870
-924 719 0.5 0.2 9070
-133 -532 0.25 0.2 7290
12 -636 0.5 0.1 9390
-986 332 0.2 0.1 6380
-237 677 -0.1 -1.0 8290
795 433 0.1 -0.25 970
669 -279 0.1 0.2 8160
-794 -551 1.0 1.0 9200
149 -108 -0.1 0.5 9210
-988 -158 -1.0 -1.0 9250
-469 226 0.1 -0.1 7420

```

**Стандартный вывод**

```
17
```

***Пример программы-решения***

Ниже представлено решение на языке Python.

```

1 import matplotlib.pyplot as plt
2 from matplotlib import patches
3
4 def display_map(figures, points, x_min, x_max, y_min, y_max):
5     plt.figure(figsize=(10, 10))
6     for fig in figures:
7         patch = None
8         if len(fig) == 3:
9             patch = patches.Circle(fig[:2], fig[2])
10        if len(fig) == 6:

```

```

11         patch = patches.Polygon(list(zip(*(fig[:,2], fig[1:,2])))
12     if len(fig) == 4:
13         patch = patches.Rectangle(fig[:,2], fig[3], fig[2])
14     if patch is None:
15         raise ValueError("Broken data file")
16     plt.gca().add_patch(patch)
17     for p in points:
18         patch = patches.Circle(p[:,2], 3, ec=(1, 0, 0))
19         plt.gca().add_patch(patch)
20     plt.xlim(x_min, x_max)
21     plt.ylim(y_min, y_max)
22     plt.show()
23
24 def move_points(points, x_min, x_max, y_min, y_max):
25     for p in points:
26         x, y, vx, vy, sp = p
27         for i in range(int(sp)):
28             x += vx
29             y += vy
30             if x >= x_max or x <= x_min:
31                 vx = -vx
32             if y >= y_max or y <= y_min:
33                 vy = -vy
34         p[0] = x
35         p[1] = y
36
37 def rect_collision(rect, point):
38     xmin, ymin, hr, wr = rect
39     x, y = point[:,2]
40     xmax = xmin + wr
41     ymax = ymin + hr
42     return xmin <= x <= xmax and ymin <= y <= ymax
43
44 def circ_collision(circ, point):
45     cx, cy, r = circ
46     x, y = point[:,2]
47     d = ((x - cx) ** 2 + (y - cy) ** 2) ** 0.5
48     return d <= r
49
50 def tri_collision(tri, point):
51     x1, y1, x2, y2, x3, y3 = tri
52     x, y = point[:,2]
53
54     area = abs((x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1))
55
56     area1 = abs((x1 - x) * (y2 - y) - (x2 - x) * (y1 - y))
57     area2 = abs((x2 - x) * (y3 - y) - (x3 - x) * (y2 - y))
58     area3 = abs((x3 - x) * (y1 - y) - (x1 - x) * (y3 - y))
59
60     return abs((area1 + area2 + area3) - area) < 1
61
62 def check_collision(points, figures):
63     nc = 0
64     while len(points) > 0:
65         point = points.pop()
66         for fig in figures:
67             if len(fig) == 3:
68                 f = circ_collision
69             elif len(fig) == 4:
70                 f = rect_collision

```

```
71         elif len(fig) == 6:
72             f = tri_collision
73         else:
74             raise ValueError("Invalid input data")
75         if f(fig, point):
76             nc += 1
77             break
78     return nc
79
80 x_min, x_max, y_min, y_max = map(int, input().split())
81 nf, np = map(int, input().split())
82
83 figures = []
84 for _ in range(nf):
85     figures.append(list(map(int, input().split())))
86
87 points = []
88 for _ in range(np):
89     points.append(list(map(float, input().split())))
90
91 move_points(points, x_min, x_max, y_min, y_max)
92 display_map(figures, points, x_min, x_max, y_min, y_max)
93 print(check_collision(points, figures))
```

#### *Задача III.1.1.4. (25 баллов)*

Написать программу для определения количества ячеек по высоте и ширине на изображении содержащим периодическую структуру (рисунок III.1.5). На изображении всегда есть ячейки и отсутствуют строки или столбцы в которых нет ни одной ячейки. Ячейки отделены друг от друга минимум на одно значение. Значения находятся в диапазоне от 0 до 255. Размер ячеек прямоугольный. Количество ячеек по высоте и ширине может быть разным.

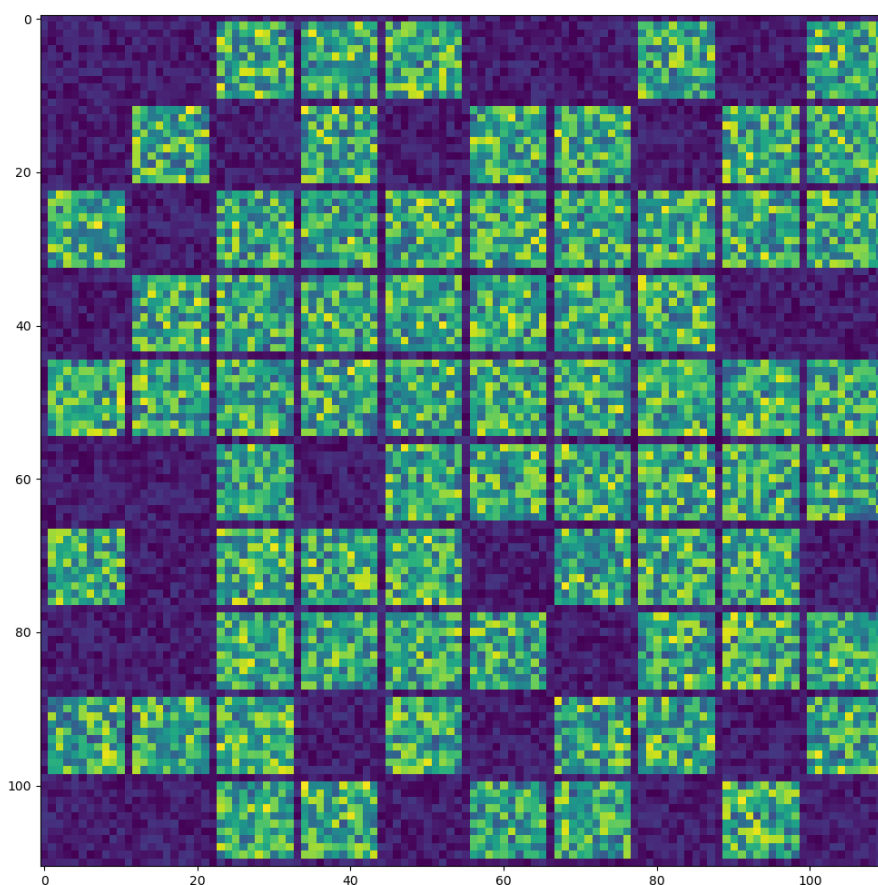


Рис. III.1.5: Пример изображения с периодической структурой

### *Формат входных данных*

В первой строке через пробел заданы высота  $h$  и ширина  $w$  матрицы. В следующих строках заданы значения матрицы по строкам и столбцам.

### *Формат выходных данных*

Количество ячеек по вертикали и горизонтали в виде двух чисел разделенных пробелом.

### *Примеры*

*Пример №1*

**Стандартный ввод**

```

111 111
30 22 18 36 29 29 20 3 23 38 28 26 19 6 13 16 9 13 35 28 34 6 9
39 10 30 22 38 16 31 38 8 22 21 38 38 16 32 29 17 35 29 14 18 37 5 89
28 11 13 33 30 16 18 38 24 2 15 21 20 28 8 10 33 12 4 2 29 38 13 119
18 3 24 27 29 28 27 35 30 6 32 3 37 19 31 23 17 27 17 3 39 24 35 228
8 31 2 27 32 2 15 27 3 1 2 30 24 34 4 23 10 30 10 14 1 33 11 196 211 75
25 34 26 3 35 21 27 29 15 22 9 28 4 11 10 22 11 24 1 36 16 1 36 86 166
31 20 8 35 32 6 34 18 38 26 36 32 28 10 37 24 28 7 17 13 21 16 5 188
30 24 38 20 21 36 38 39 8 12 13 5 26 10 35 3 34 30 30 27 12 36 10 195
35 31 20 33 31 8 5 3 28 31 31 13 35 25 35 26 8 24 24 35 32 21 37 63 130

```

В таблице приведена часть входных данных.

**Стандартный вывод**

```
10 10
```

**Пример программы-решения**

Ниже представлено решение на языке Python.

```

1  def get_grid_level(image):
2  values = []
3  for row in image:
4      values.extend(row)
5  levels = list(set(sorted(values)))
6  diff = []
7  for i in range(1, len(levels)):
8      diff.append(levels[i] - levels[i-1])
9  index = diff.index(max(diff))
10 return levels[index]
11
12 def binarize_image(image):
13 level = get_grid_level(image)
14 for row in image:
15     for i in range(len(row)):
16         if row[i] > level:
17             row[i] = 1
18         else:
19             row[i] = 0
20
21 def mean_row_col(image):
22 col = []
23 for r in image:
24     col.append(sum(r))
25
26 row = image[0].copy()
27 for r in image[1:]:
28     for i in range(len(r)):
29         row[i] += r[i]
30
31 return col, row
32
33 def count_step(line):
34 n = 0
35 for i in range(1, len(line)):
36     if line[i] > 0 and line[i-1] == 0:
37         n += 1

```

```
38     return n
39
40 h, w = map(int, input().split())
41 image = []
42
43 for i in range(h):
44     line = map(int, input().split())
45     image.append(list(line))
46
47 binarize_image(image)
48 col, row = mean_row_col(image)
49
50 nc = count_step(col)
51 nr = count_step(row)
52
53 print(nc, nr)
```

### *Задача III.1.1.5. (20 баллов)*

Важной задачей в виртуальной реальности является позиционирование объектов. В некоторых случаях необходимо, чтобы виртуальный объект следил за пользователем и был повернут к нему. Задачу можно свести к двумерному случаю, когда задача рассматривается на плоскости. Таким образом необходимо написать программу, которая определит на какой угол должны быть повернуты стрелки, так чтобы все смотрели на объект. Стрелки могут быть расположены с любой стороны от объекта. Необходимо определить такой угол при котором каждая стрелка будет повернута в сторону объекта, например круга (рисунок III.1.6). Угол в интервале от 0 до 359 (рисунок III.1.7).



Рис. III.1.6: Пример изображения с повернутыми стрелками, так чтобы они смотрели на круг

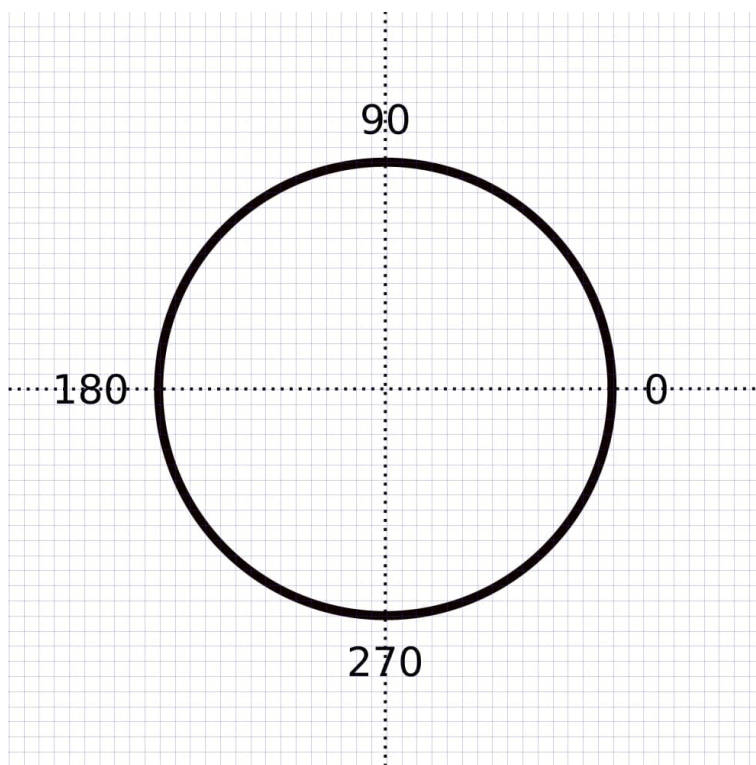


Рис. III.1.7: Координатная система

**Формат входных данных**

В первой строке заданы координаты объекта на который должны быть направлены стрелки. В следующей строке одним числом задано количество стрелок. Во всех следующих строках идут координаты  $x$  и  $y$  стрелок на плоскости.

**Формат выходных данных**

Углы в диапазоне от 0 до 359 в порядке возрастания. Каждый угол на новой строке. Без дробной части.

**Примеры***Пример №1*

Стандартный ввод	
0	0
4	
2	2
1	-1
-3	3
-4	-4
Стандартный вывод	
45	
135	
225	
315	

### Пример программы-решения

Ниже представлено решение на языке Python.

```

1 import math
2
3 def _deg(x, y):
4     angle = math.degrees(math.atan2(y, x))
5     if angle < 0:
6         angle = 360 + angle
7     return int(angle)
8
9 def calc_deg(xc, yc, points):
10    degs = []
11    for p in points:
12        x, y = p
13        xt = xc - x
14        yt = yc - y
15        degs.append(_deg(xt, yt))
16    return sorted(degs)
17
18 xc, yc = map(int, input().split())
19 n = int(input())
20
21 points = []
22 for _ in range(n):
23    points.append(list(map(int, input().split())))
24
25 for deg in calc_deg(xc, yc, points):
26    print(deg)

```

## Математика. 8-9 класс

### Задача III.1.2.1. (10 баллов)

В течение сета в AR-игре пользователю пришел счет за услуги, оказанные ему дружественным игроком: за услуги портного 9 гроутов 3 пенса; за услуги зеленщика 5 гроутов 2 пенса; за ремонт крыльца 8 гроутов 3 пенса; за услуги кузнеца 6 гроутов 3 пенса; за покупку пшеницы 2 гроута 1 пенс и за покупку овса — 2 гроута 2 пенса. Всего к оплате было затребовано 2 кроны 5 гроутов и 2 пенса.

Определите, сколько пенсов в гроуте и гроутов в кроне, если известно, что число гроутов в кроне и число пенсов в гроуте являются взаимно простыми натуральными числами. В ответ записать сумму, затребованную к оплате в пенсах.

### Решение

Всего в столбце Пенсы 14 пенсов, в столбце Гроуты - 32 гроута.

2 кроны 5 гроутов 2 пенса = 32 гроутам 14 пенсам;

2 кроны = 27 гроутов 12 пенсов (1).

Очевидно, что в сумме пенсов из каждой строки содержится целое число гроутов, при этом остаток, содержащийся в строке "Итого", является остатком от общего



числа (см. таблицу).

Услуга	Кроны	Гроуты	Пенсы
за услуги портного		9	3
за услуги зеленщика		5	2
за ремонт крыльца		8	3
за услуги кузнеца		6	3
за покупку пшеницы		2	1
за покупку овса		2	2
<b>Итого</b>	2	5	2

Следовательно, в  $14 - 2 = 12$  пенсах - целое количество гроутов. Очевидно, что в одном гроуте больше, чем 3 пенса (поскольку такое количество пенсов указано в таблице). Проверим все возможные варианты с числами, которым кратно число 12:

1. Пусть в одном гроуте 12 пенсов. Следовательно по (1), в 2 кронах  $27 \cdot 12 + 12 = 168$  пенсов, в 1 кроне 14 пенсов или  $\frac{14}{12}$  гроутов. Условия задачи не выполняются.
2. Пусть в одном гроуте 6 пенсов. Следовательно по (1), в 2 кронах  $27 \cdot 6 + 12 = 174$  пенса, в 1 кроне 87 пенсов или  $\frac{87}{6}$  гроутов. Условия задачи не выполняются.
3. Пусть в одном гроуте 4 пенса. Следовательно по (1), в 2 кронах  $27 \cdot 4 + 12 = 120$  пенсов, в 1 кроне 60 пенсов или 15 гроутов. Условия задачи полностью выполнены.

Тогда сумма заказа будет равна  $32 \cdot 4 + 14 = 142$  пенсам.

**Ответ:** 142.

### **Задача III.1.2.2. (15 баллов)**

В одном из авторизованных магазинов по продаже очков Microsoft Hololens 2 было замечено, что в 17% покупатели носят очки или имеют некоторые проблемы со зрением, из-за чего обязательно сразу же оформляют заказ на корректирующие линзы, подходящие для данной модели. Какова вероятность того что двое из семерых пришедших в течение одного дня покупателей будут заказывать корректирующие линзы? Ответ округлить до тысячных.

#### **Решение**

Обозначим вероятность того, что пришедший покупатель имеет проблемы со зрением как  $p = 0,17$ , а вероятность того что проблем со зрением нет как  $q = 1 - p = 0,83$ . Вероятность того, что двое из семи покупателей будут иметь проблемы со зрением равна  $p^2 \cdot q^5 = \frac{1\,138\,382\,745\,827}{100\,000\,000\,000\,000}$ . Определим количество вариантов возникновения таких ситуаций:  $C_7^2 = \frac{7!}{2!5!} = 21$ . Таким образом, вероятность того, что покупатели будут заказывать корректирующие линзы равна  $21 \cdot \frac{1\,138\,382\,745\,827}{100\,000\,000\,000\,000} = \frac{23\,906\,037\,662\,367}{100\,000\,000\,000\,000} \approx 0,239$

**Ответ:** 0,239.

### Задача III.1.2.3. (20 баллов)

Компания выпустила новую версию AR-очков, с функционалом измерения длины объектов в реальном мире по принципу рулетки или линейки. Пользователь  $N$ , купивший такие очки, решил замерить при полете на планере периметр своего участка. Какую величину периметра (в км) увидит пользователь, если известно, что его земляной надел имеет форму треугольника, построенного на основаниях высот остроугольного треугольного поля со сторонами 5, 6 и 7 км. Ответ записать в виде неправильной несократимой дроби в виде "(числитель; знаменатель)".

#### Решение

Треугольники  $\triangle ABC$  и  $\triangle AFE$  подобны (см. чертеж).

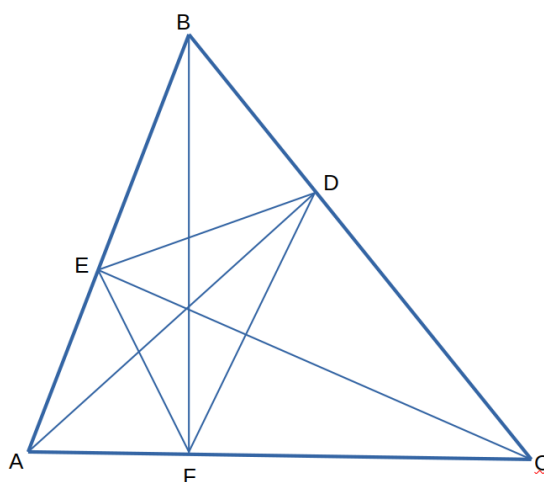


Рис. III.1.8: Чертеж к задаче

Тогда  $\frac{FE}{BC} = \frac{AF}{AB}$  и, следовательно,  $AF = \frac{AB}{BC} \cdot FE = AB \cdot \cos \angle A$  и  $FE = \frac{BC}{AB} \cdot AF$ .

Из теоремы косинусов для  $\angle A$  треугольника  $\triangle ABC$ :

$$BC^2 = AC^2 + AB^2 - 2AC \cdot AB \cdot \cos \angle A = AC^2 + AB^2 - 2AF \cdot BC.$$

$$\text{Отсюда } AF = \frac{AC^2 + AB^2 - BC^2}{2AC} = 5 \text{ и } FE = \frac{BC(AC^2 + AB^2 - BC^2)}{2AC \cdot AB} = \frac{25}{7}.$$

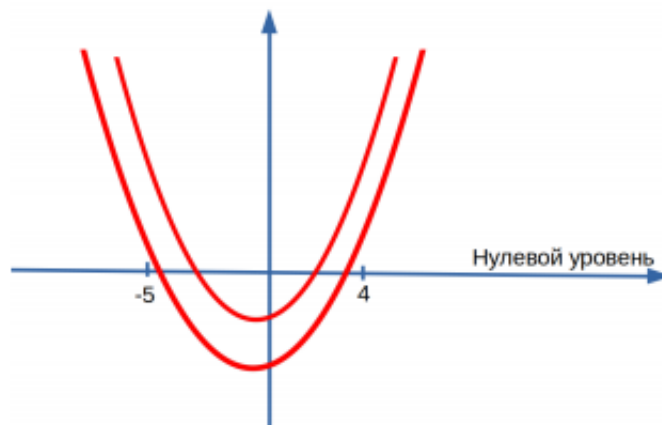
$$\text{Аналогично находим } ED = \frac{AC(BC^2 + AB^2 - AC^2)}{2BC \cdot AB} = \frac{3 \cdot 38}{5 \cdot 7} \text{ и } FD = \frac{AB(BC^2 + AC^2 - AB^2)}{2AC \cdot BC} = \frac{7}{5}.$$

$$\text{Периметр треугольника } \triangle DEF \text{ равен } FE + FD + DE = \frac{288}{35}.$$

**Ответ:** (8855; 1026).

### Задача III.1.2.4. (20 баллов)

Для отработки маневра с нырянием и движением вдоль некоторой поверхности аквалангист движется по параболической траектории  $y(x) = 5x^2 - (4m + 7)x - 2m - 3$ . При каких значениях параметра  $m$  ему удастся пересечь нулевую отметку два раза так, чтобы оба прохождения через эту отметку были заключены между  $-5$  и  $4$  метрами (см. чертеж). Ответ записать двойным неравенством с неправильными несократимыми дробями.



### Решение

Чтобы у аквалангиста было и погружение, и подъем и, соответственно, два пересечения нулевой отметки, необходимо, чтобы уравнение имело два корня, и, следовательно, дискриминант был положителен. Так,  $D = (4m + 7)^2 + 20(2m + 3) = 16m^2 + 96m + 109 > 0$  при любом  $m$ .

Пусть  $x_1$  и  $x_2$  - корни данного уравнения, тогда по условию,  $-5 < x_1 < 4$  и  $-5 < x_2 < 4$ .

Следовательно,  $f(x) = 5x^2 - (4m + 7)x - 2m - 3$  квадратичная функция, а ее графиком является парабола с ветвями, направленными вверх. Поскольку эта функция имеет два корня, находящихся в заданном промежутке, то, очевидно, что  $f(-5) > 0$  и  $f(4) > 0$ . Таким образом, можно получить систему неравенств:

$$\begin{cases} 5(-5)^2 - (4m + 7) \cdot (-5) - 2m - 3 > 0, \\ 5(4)^2 - (4m + 7) \cdot 4 - 2m - 3 > 0, \end{cases} \Rightarrow \begin{cases} 157 + 18m > 0, \\ 49 - 18m > 0, \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} m > -\frac{157}{18}, \\ m < \frac{49}{18}. \end{cases}$$

**Ответ:**  $-\frac{157}{18} < m < \frac{49}{18}$ .

### Задача III.1.2.5. (25 баллов)

Решить систему уравнений:

$$\begin{cases} \sqrt{2x + y} + \sqrt{y + 3z} = 7, \\ \sqrt{y + 3z} + \sqrt{2x + 3z} = 4, \\ \sqrt{2x + y} + \sqrt{2x + 3z} = 5. \end{cases}$$

Ответ записать в виде  $(x; y; z)$ .

### Решение

Сложим все три уравнения в системе получим, что  $\sqrt{2x + y} + \sqrt{y + 3z} + \sqrt{2x + 3z} = 8$ . Далее отнимем из получившейся суммы каждое из уравнений:

$$\begin{cases} \sqrt{2x+y} = 4, \\ \sqrt{y+3z} = 3, \\ \sqrt{2x+3z} = 1, \end{cases} \Rightarrow \begin{cases} 2x+y = 16, \\ y+3z = 9, \\ 2x+3z = 1. \end{cases} \Rightarrow \begin{cases} x = 2, \\ y = 12, \\ z = -1. \end{cases}$$

Ответ: (2;12;-1).

## Математика. 10-11 класс

### Задача III.1.3.1. (15 баллов)

Инженеры, участвующие в разработке AR-очков, запрограммировали появление на рабочем поле сообщений, которые можно перемещать вправо, влево, вверх и вниз на 7, 10, 11 и 8 пикселей, соответственно. За какое минимальное количество ходов пользователь сможет приподнять окошко на 3 пикселя и сместить вправо на 2 пикселя?

#### Решение

Пусть  $x_1$ ,  $y_1$ ,  $x_2$  и  $y_2$  - количество перемещений вправо, влево, вверх и вниз соответственно. Тогда можно составить два диофантова уравнения:

$$\begin{cases} 7x_1 - 10y_1 = 2, \\ 11x_2 - 8y_2 = 3. \end{cases}$$

Решим данные диофантовы уравнения (поскольку число ходов должно быть целочисленными) с помощью цепных дробей:

$$\begin{cases} \frac{7}{10} = 0 + \frac{1}{1 + \frac{1}{2 + \frac{1}{3}}}, \\ \frac{11}{8} = 1 + \frac{1}{2 + \frac{1}{1 + \frac{1}{2}}}. \end{cases}$$

Далее заменим в каждой из дробей последнюю дробь на  $\frac{1}{1}$  и вычислим значения получившихся дробей:

$$\begin{cases} 0 + \frac{1}{1 + \frac{1}{2 + \frac{1}{1}}} = \frac{3}{4}, \\ 1 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1}}} = \frac{7}{5}. \end{cases}$$

Разница между исходными и получившимися дробями даст нам две пары частных решений данных диофантовых уравнений:

$$\begin{cases} \frac{7}{10} - \frac{3}{4} = \frac{7 \cdot 4 - 10 \cdot 3}{4 \cdot 10} = \frac{-2}{4 \cdot 10}, \\ \frac{11}{8} - \frac{7}{5} = \frac{11 \cdot 5 - 8 \cdot 7}{5 \cdot 8} = \frac{-1}{5 \cdot 8}. \end{cases}$$

При сравнении числителей видно, что

$$\begin{cases} 7 \cdot 4 - 10 \cdot 3 = -2, | \cdot (-1) \\ 11 \cdot 5 - 8 \cdot 7 = -1, | \cdot (-3) \end{cases} \Rightarrow \begin{cases} 7 \cdot (-4) - 10 \cdot (-3) = 2, \\ 11 \cdot (-15) - 8 \cdot (-21) = 3. \end{cases}$$

Таким образом, получим два частных решения:

$$\begin{cases} x_1 = -4 + 10t_1, \\ y_1 = -3 + 7t_1, \\ x_2 = -15 + 8t_2, \\ y_2 = -21 + 11t_2, \end{cases}$$

Следовательно, наименьшими положительными значениями

$$\begin{cases} x_1 = -4 + 10 \cdot 1 = 6, \\ y_1 = -3 + 7 \cdot 1 = 4, \\ x_2 = -15 + 8 \cdot 2 = 1, \\ y_2 = -21 + 11 \cdot 2 = 1. \end{cases}$$

Итого будет 12 перемещений.

**Ответ:** 26.

### **Задача III.1.3.2. (20 баллов)**

При лингвистическом анализе одного из диалектов языка ротокас с 12-тибуквенным алфавитом разработчиком приложения были изучены слова, состоящие из трех, четырех и пяти букв. Оказалось, что слова, имеющие хоть какой-то смысл, составляют 12,5% от возможного количества сочетаний букв. Сколько слов, имеющих смысл, есть в словарном запасе носителя среди изучаемых слов?

#### **Решение**

Всего в таком языке может быть  $12^3 + 12^4 + 12^5 = 271\,296$  слов. Следовательно, имеющих слов в словарном запасе составляет  $271\,296 \cdot 0.125 = 33\,912$  слов.

**Ответ:** 33 912.

### **Задача III.1.3.3. (20 баллов)**

Для игры в квест необходимо установить RFID-метку в фигуру полученную из двух равных кубов с диагональю, лежащей на одной прямой, причем вершина одного куба совпадает с центром второго куба, повернутого вокруг общей диагонали на  $60^\circ$  по отношению к первому. Известно, что метка должна быть установлена внутри общей части этих кубов. Найти объем (в  $\text{см}^3$ ) этой части если известно, что сторона куба равна 15 см. Ответ округлить до десятых.

#### **Решение**

Пусть первоначальный угол поворота был равен  $0^\circ$ . Если провести сечение через центр общей части двух кубов перпендикулярно диагонали, то получится правильный шестиугольник. После поворота на  $60^\circ$  этот шестиугольник совместится с самим собой из-за того что ребра кубов пересекаются.

Искомая часть состоит из двух одинаковых правильных треугольных пирамид с вершинами в точках  $O$  и  $O_1$ . Поскольку все углы при вершине - прямые, то объем каждой пирамиды равен  $\frac{OA^3}{6} = \frac{b^3}{6}$ . Высота каждой из пирамид  $h = \frac{OO_1}{2} = \frac{AB\sqrt{3}}{4} = \frac{a\sqrt{3}}{4}$ . С другой стороны  $V_1 = \frac{1}{3}S_{\text{осн}}h$ , где  $S_{\text{осн}} = \frac{(b\sqrt{2})^2\sqrt{3}}{4} = \frac{b^2\sqrt{3}}{2}$ . Получается, что  $\frac{b^3}{6} = \frac{1}{3}\frac{b^2\sqrt{3}}{2}h$ , откуда  $h = \frac{b\sqrt{3}}{3}$ . Следовательно  $\frac{b\sqrt{3}}{3} = \frac{a\sqrt{3}}{4}$ ,  $b = \frac{3}{4}a$ . Таким образом, искомый объем равен  $V = 2V_1 = \frac{b^3}{3} = \frac{9a^3}{64} = 474.6$ .

**Ответ:** 474, 6.

### Задача III.1.3.4. (20 баллов)

При построении виртуального пространства для игры «Поймай Пасхального Кролика» в программу было заложено, что Кролик может двигаться в области, описанной условием  $|2y + 3x - 2| + |3x + 6| < 6$ . Каким должен быть минимальный радиус идеально круглой ловушки, чтобы Кролик сразу был пойман?

#### Решение

Построим фигуру, заданную неравенством, рассмотрев для этого четыре варианта раскрытия модулей.

$$1) \begin{cases} 2y + 3x - 2 \geq 0, \\ 3x + 6 \geq 0, \\ 2y + 3x - 2 + 3x + 6 < 6, \end{cases} \Rightarrow \begin{cases} y \geq -1.5x + 1, \\ x \geq -2, \\ y < -3x + 1. \end{cases}$$

Данная система неравенств соответствует внутренности треугольника с вершинами в точках  $(0; 1)$ ,  $(-2; 4)$ ,  $(-2; 7)$ . Аналогично необходимо найти фигуры для остальных случаев:

$$2) \begin{cases} 2y + 3x - 2 \leq 0 \\ 3x + 6 \geq 0 \\ -(2y + 3x - 2) + 3x + 6 < 6 \end{cases} \quad 3) \begin{cases} 2y + 3x - 2 \geq 0, \\ 3x + 6 \leq 0, \\ 2y + 3x - 2 - (3x + 6) < 6, \end{cases}$$

$$4) \begin{cases} 2y + 3x - 2 \leq 0, \\ 3x + 6 \leq 0, \\ -(2y + 3x - 2) - (3x + 6) < 6. \end{cases}$$

В результате получим параллелограмм с вершинами в точках  $(0; 1)$ ,  $(-2; 1)$ ,  $(-4; 7)$ ,  $(-2; 7)$ . Центр окружности будет располагаться в точке  $(-2; 4)$ , а ее радиус равен половине большей диагонали  $R^2 = (0 + 2)^2 + (4 - 1)^2 = 13$ . Следовательно, радиус равен  $\sqrt{13}$ .

**Ответ:** 13.

### Задача III.1.3.5. (20 баллов)

Решить неравенство:

$$\frac{\log_5(x^2 - 4x - 11)^2 - \log_{11}(x^2 - 4x - 11)^3}{2 - 5x - 3x^2} \leq 0.$$

### Решение

Обозначим область допустимых значений:

$$\begin{cases} 2x^2 - 4x - 11 > 0, \\ 2 - 5x - 3x^2 \neq 0, \end{cases} \Rightarrow \begin{cases} x > 2 + \sqrt{15}, \\ x < 2 - \sqrt{15}, \\ x_3 \neq -2, x_4 \neq \frac{1}{3}. \end{cases}$$

Далее можно преобразовать неравенство:

$$\frac{2 \log_5 (x^2 - 4x - 11) - 3 \log_5 (x^2 - 4x - 11) / \log_5 11}{2 - 5x - 3x^2} \leq 0,$$

$$\frac{(2 \log_5 11 - 3) \log_5 (x^2 - 4x - 11)}{\log_5 11 \cdot (2 - 5x - 3x^2)} \leq 0.$$

Поскольку  $\log_5 11 > 0$ , а  $2 \log_5 11 - 3 = \log_5 \frac{11^2}{5^3} < 0$ , то

$$\frac{\log_5 (x^2 - 4x - 11)}{(3x^2 + 5x - 2)} \leq 0.$$

Возможны два случая:

- 1)  $\begin{cases} \log_5 (x^2 - 4x - 11) \geq 0, \\ 3x^2 + 5x - 2 < 0, \end{cases} \Rightarrow \begin{cases} x^2 - 4x - 12 \geq 0, \\ 3x^2 + 5x - 2 < 0, \end{cases} \Rightarrow x \in \emptyset.$
- 2)  $\begin{cases} \log_5 (x^2 - 4x - 11) \leq 0, \\ 3x^2 + 5x - 2 > 0, \end{cases} \Rightarrow \begin{cases} x^2 - 4x - 12 \leq 0, \\ 3x^2 + 5x - 2 > 0, \end{cases} \Rightarrow x \in (\frac{1}{3}; 6].$

С учетом ОДЗ  $x \in (2 + \sqrt{15}; 6].$

**Ответ:**  $(2 + \sqrt{15}; 6].$