

# Яндекс. Тренировки по алгоритмам июнь 2021, занятие 3

## J. Пробежки по Манхэттену

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дороги Нью-Манхэттена устроены следующим образом. С юга на север через каждые сто метров проходит авеню, с запада на восток через каждые сто метров проходит улица. Авеню и улицы нумеруются целыми числами. Меньшие номера соответствуют западным авеню и южным улицам. Таким образом, можно построить прямоугольную систему координат так, чтобы точка  $(x, y)$  лежала на пересечении  $x$ -ой авеню и  $y$ -ой улицы. Легко заметить, что для того, чтобы в Нью-Манхэттене дойти от точки  $(x_1, y_1)$  до точки  $(x_2, y_2)$  нужно пройти  $|x_2 - x_1| + |y_2 - y_1|$  кварталов. Эта величина называется манхэттенским расстоянием между точками  $(x_1, y_1)$  и  $(x_2, y_2)$ .

Миша живет в Нью-Манхэттене и каждое утро делает пробежку по городу. Он выбегает из своего дома, который находится в точке  $(0, 0)$  и бежит по случайному маршруту. Каждую минуту Миша либо остается на том же перекрестке, что и минуту назад, или перемещается на один квартал в любом направлении. Чтобы не заблудиться Миша берет с собой навигатор, который каждые  $t$  минут говорит Мише, в какой точке он находится. К сожалению, навигатор показывает не точное положение Миши, он может показать любую из точек, манхэттенское расстояние от которых до Миши не превышает  $d$ .

Через  $t \times n$  минут от начала пробежки, получив  $n$ -е сообщение от навигатора, Миша решил, что пора бежать домой. Для этого он хочет понять, в каких точках он может находиться. Помогите Мише сделать это.

### Формат ввода

Первая строка входного файла содержит числа  $t$ ,  $d$  и  $n$  ( $1 \leq t \leq 100$ ,  $1 \leq d \leq 100$ ,  $1 \leq n \leq 100$ ).

Далее  $n$  строк описывают данные, полученные от навигатора. Строка номер  $i$  содержит числа  $x_i$  и  $y_i$  — данные, полученные от навигатора через  $t_i$  минут от начала пробежки.

### Формат вывода

В первой строке выходного файла выведите число  $m$  — число точек, в которых может находиться Миша. Далее выведите  $m$  пар чисел — координаты точек. Точки можно вывести в произвольном порядке.

Гарантируется, что навигатор исправен и что существует по крайней мере одна точка, в которой может находиться Миша.

#### Пример 1

Ввод

Вывод

2 1 5  
0 1  
-2 1  
-2 3  
0 3  
2 5

2  
1 5  
2 4

#### Пример 2

Ввод

1 1 1  
0 0

Вывод

5  
-1 0  
0 -1  
0 0  
0 1  
1 0

### Пример 3

Ввод

1 10 1  
0 0

Вывод

5  
-1 0  
0 -1  
0 0  
0 1  
1 0

Язык Python 3.12.1

Набрать здесь

Отправить файл

```
1 def extend(rect, d):
2     min_plus, max_plus, min_minus, max_minus = rect
3     return [min_plus - d, max_plus + d, min_minus - d, max_minus + d]
4
5 def intersect(rect1, rect2):
6     ans = [max(rect1[0], rect2[0]), min(rect1[1], rect2[1]), max(rect1[2], rect2[2]), min(rect1[3], rect2[3])]
7     return ans
8
9
10 t, d, n = map(int, input().split())
11
12 pos_rect = (0, 0, 0, 0) # min(x+y), max(x+y), min(x-y), max(x-y)
13
14 for _ in range(n):
15     pos_rect = extend(pos_rect, t)
16     nav_x, nav_y = map(int, input().split())
17     nav_rect = extend((nav_x + nav_y, nav_x + nav_y, nav_x - nav_y, nav_x - nav_y), d)
18     pos_rect = intersect(pos_rect, nav_rect)
19
20 probab_points = []
21 for x_plus_y in range(pos_rect[0], pos_rect[1] + 1):
22     for x_minus_y in range(pos_rect[2], pos_rect[3] + 1):
23         if (x_plus_y + x_minus_y) % 2 == 0:
24             x = (x_plus_y + x_minus_y) // 2
25             y = x_plus_y - x
26             probab_points.append((x, y))
27
28
29 print(len(probab_points))
30 print('\n'.join(f'{str(a)}+' +str(b) for a, b in probab_points))
31
```

Отправить

Предыдущая