

Подзадача №1: необходимо написать программу на вход которой будет подаваться массив с рандомно сгенерированными числами. Из массива изначально необходимо вывести все четные числа в порядке возрастания, а затем все нечетные числа в порядке убывания их индексов.

```
import random

iter = 0
list = []

while iter < 10:
    list.append(random.randint(0, 100))
    iter += 1

print(f'Массив: {list}')

iter = 0

print('Чётные числа в порядке возрастания:')

while iter < len(list):
    if list[iter] % 2 == 0:
        print(list[iter])
        iter += 1

iter = len(list) - 1

print('Нечётные числа в порядке убывания:')

while iter >= 0:
    if list[iter] % 2 != 0:
        print(list[iter])
        iter -= 1
```

Протокол:

Массив: [39, 42, 7, 40, 87, 36, 14, 54, 45, 85]

Чётные числа в порядке возрастания:

42

40

36

14

54

Нечётные числа в порядке убывания:

85

45

87

7

39

Process finished with exit code 0

Массив: [8, 12, 30, 77, 25, 29, 21, 70, 41, 28]

Чётные числа в порядке возрастания:

8

12

30

70

28

Нечётные числа в порядке убывания:

41

21

29

25

77

Process finished with exit code 0

Подзадача №2: Необходимо написать программу которая будет находить участки в списке которые монотонно убывают

```

import random # Импортируем библиотеку random
def program():
    try:
        lst = [random.randint(1, 100) for el in range(int(input('Введите размер списка: ')))]
        # Заполняем список размера N, случайными значениями
        print(lst) # Выводим созданный список на экран
        result = 0
        count = 0
        for j in range(len(lst) - 2):
            if lst[j + 2] < lst[j + 1] < lst[j]:
                count += 1
            elif count >= 1 and lst[j + 1] < lst[j + 2]: # Поиск монотонных возрастаний
                result += 1
                count = 0
            if lst[-1] < lst[-2] < lst[-3]:
                result += 1 # Запись кол-ва элементов возрастания
        print(f'Ответ: {result}') # Вывод результата

    except ValueError:
        print("Ошибка ввода") # Оповещение об ошибке
        program() # Повторный вызов функции из-за ошибки

program()

```

Протокол:

Введите размер списка: 10
 [41, 34, 69, 29, 93, 57, 4, 26, 71, 5]
 Ответ: 1

Process finished with exit code 0

Введите размер списка: sd
 Ошибка ввода

Подзадача №3: Необходимо написать программу которая будет искать ближайшую к «В» точку из списка

```

from math import sqrt
from random import randint

def program():
    try:
        RANGE = int(input('Введите число точек: '))

        x = []
        y = []

        answers = []

        for i in range(1, RANGE + 1): #Рандомно генерируются координаты точек
            x.append(randint(1, 10))
            y.append(randint(1, 10))

        print(f'x:{x}\ny:{y}')

        b = [randint(1, 10), randint(1, 10)] #Рандомно генерируется точка B

        i = 0
        j = 0

        while i <= len(x)-1 and j <= len(y)-1:
            r = sqrt(((b[0] - x[i]) ** 2) + ((b[1] - y[j]) ** 2)) #формула по которой вычисляется
расстояние
            answers.append(r) #Все расстояние загружаются в массив
            i += 1
            j += 1

        """
        Ниже из всех расстояний выбирается наименьшее и находится индекс этого
значения в массиве answers
с помощью которого вычисляются координаты точек из которых было вычисленно
это расстояние
        """

        # r = sqrt(((b[0] - x[0])**2) + ((b[1] - y[0])**2))
        print(f'Все расстояния: {answers}')
        indexOfAnswers = answers.index(min(answers))

        print('\n\n')

        print(f'Точка с координатами ({x[indexOfAnswers]}; {y[indexOfAnswers]}) является
ближайшей к точке B с координатами ({b[0]}; {b[1]})\n'
            f'Расстояние до точки: {min(answers)}')

```

```
except ValueError:  
    print('Ошибка ввода')  
    program()  
  
program()
```

Протокол:

Введите число точек: 10

х:[2, 2, 8, 1, 2, 6, 2, 2, 8, 5]

у:[3, 5, 1, 4, 10, 4, 9, 3, 10, 10]

Все расстояния: [2.23606797749979, 1.0,
6.4031242374328485, 2.23606797749979,
5.0990195135927845, 3.1622776601683795,
4.123105625617661, 2.23606797749979,
7.0710678118654755, 5.385164807134504]

Точка с координатами (2; 5) является ближайшей к
точке В с координатами (3; 5)

Расстояние до точки: 1.0

Process finished with exit code 0

Введите число точек: 5

х:[10, 6, 6, 2, 8]

у:[3, 6, 5, 1, 5]

Все расстояния: [9.219544457292887,
4.242640687119285, 5.0, 8.06225774829855,

6.4031242374328485]

Точка с координатами (6; 6) является ближайшей к
точке В с координатами (3; 9)

Расстояние до точки: 4.242640687119285

Process finished with exit code 0

Введите число точек: dsdsd

Ошибка ввода