

## Отчёт

### Практическое занятие № 6

#### Тема:

Наименование практического занятия: составление программ со списками в IDE PyCharm Community.

#### Цель:

Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

#### Постановка задачи.

Дан целочисленный список размера 10. Вывести вначале все содержащиеся в данном списке четные числа в порядке возрастания их индексов, а затем — все нечетные числа в порядке убывания их индексов.

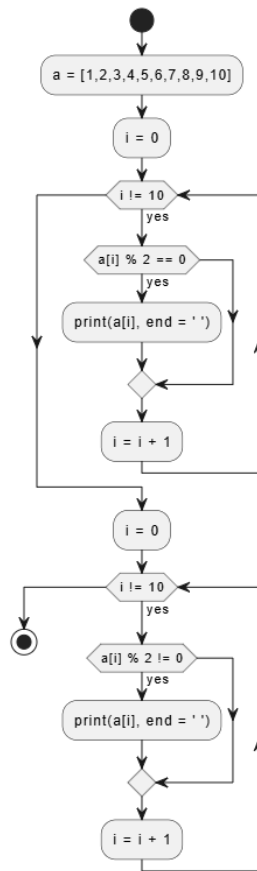
Дан список размера N. Найти количество участков, на которых его элементы монотонно убывают.

Найти точку из множества A, наиболее близкую к точке B.

Расстояние R между точками с координатами (x1, y1) и (x2, y2) вычисляется по формуле:  $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$ .

**Тип алгоритма:** Разветвлённый, Циклический

### Блок-схема алгоритма 1:



### Текст программы 1:

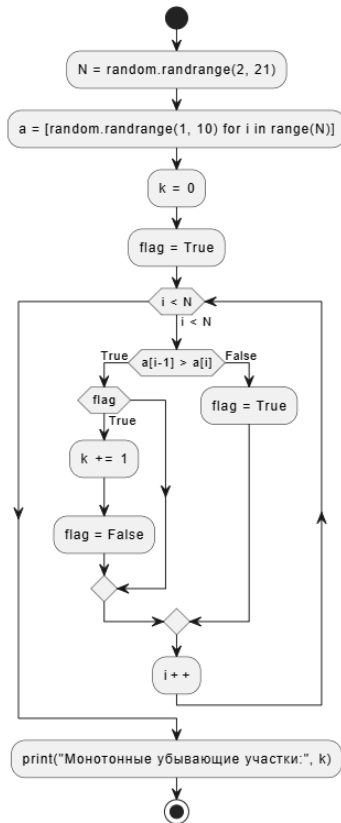
```
a = [1,2,3,4,5,6,7,8,9,10]
i = 0
while i != 10:
    if a[i] % 2 == 0:
        print(a[i], end = ' ')
    i += 1
i = 0
while i != 10:
    if (a[i] % 2 != 0):
        print(a[i], end = ' ')
    i += 1
```

### Протокол работы программы 1:

2 4 6 8 10 1 3 5 7 9

Process finished with exit code 0

## Блок-схема алгоритма 2:



## Текст программы 2:

```
import random
```

```
N = random.randrange(2, 21) # N = 15
a = [random.randrange(1, 10) for i in range(N)]
print("N = ", N)
print("Список: ", a)
k = 0
flag = True
for i in range(1, N):
    if a[i-1] > a[i]:
        if flag:
            k += 1
            flag = False
        else:
            flag = True
print("Монотонные убывающие участки:", k)
```

## Протокол работы программы 2:

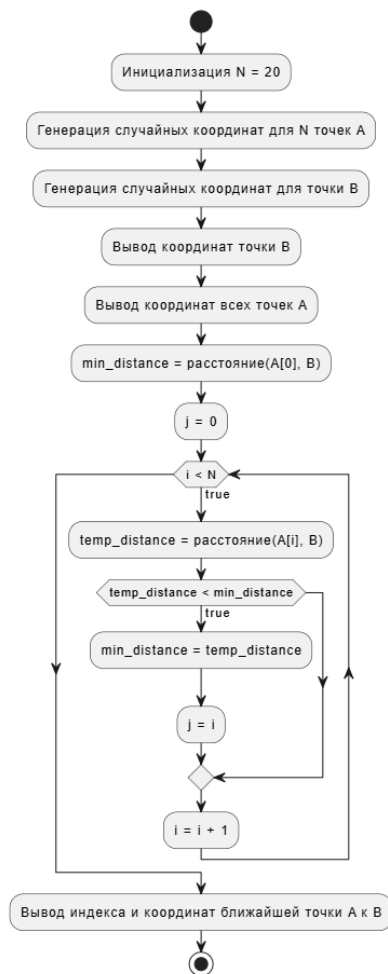
N = 15

Список: [4, 6, 9, 4, 9, 8, 4, 6, 7, 8, 3, 9, 6, 1, 7]

Монотонные убывающие участки: 4

Process finished with exit code 0

### Блок-схема алгоритма 3:



### Текст программы 3:

```
import random
import math

N = 20
class Tochka:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
def main():
    random.seed()
    A = [Tochka(random.randint(0, 49), random.randint(0, 49)) for _ in range(N)]
    b = Tochka(random.randint(0, 49), random.randint(0, 49))
    print(f'Tochka B ({b.x}, {b.y})\n')
    for i in range(N):
        print(f'Tochka {i + 1} ({A[i].x}, {A[i].y})")
    min_distance = math.sqrt((A[0].x - b.x) ** 2 + (A[0].y - b.y) ** 2)
    j = 0
    for i in range(1, N):
        temp_distance = math.sqrt((A[i].x - b.x) ** 2 + (A[i].y - b.y) ** 2)
        if temp_distance < min_distance:
            min_distance = temp_distance
```

```
j = i
print(f'Наиближайшая точка A к точке B - Точка {j + 1} ({A[j].x}, {A[j].y})')
main()
```

### **Протокол работы программы 3:**

Точка B (1, 22)

Точка 1 (44, 31)

Точка 2 (25, 20)

Точка 3 (4, 22)

Точка 4 (16, 36)

Точка 5 (45, 22)

Точка 6 (20, 23)

Точка 7 (12, 48);;

Точка 8 (26, 25)

Точка 9 (46, 18)

Точка 10 (12, 46)

Точка 11 (3, 47)

Точка 12 (49, 1)

Точка 13 (33, 45)

Точка 14 (30, 8)

Точка 15 (5, 16)

Точка 16 (35, 31)

Точка 17 (13, 29)

Точка 18 (47, 12)

Точка 19 (7, 17)

Точка 20 (40, 28)

Наиближайшая точка A к точке B - Точка 3 (4, 22)

Process finished with exit code 0

### **Вывод:**

В процессе работы были использованы процессы:if,while,else,import,def

Данный документ и код программы выложены на Github