

## Практическое занятие №6

**Тема:** Составление программ со списками в IDE PyCharm Community.

**Цель:** Закрепить усвоенные знания с IDE PyCharm Community, составление программ со списками.

### Постановка задачи:

1. 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.
2. Дан список размера N. Найти количество участков, на которых его элементы монотонно убывают.
3. #Дано множество A из N точек на плоскости и точка B (точки заданы своими координатами x,y).  
#Найти точку из множества A, наиболее близкую у 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:** Списки.

### Текст программы:

1. #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.  
import random  
spisok = []  
numbers = [random.randint(0, 999) for \_ in range(11)]  
print(numbers)  
  
even\_numbers = [numbers[i] for i in range(len(numbers)) if numbers[i] % 2 == 0]

```
odd_numbers = [numbers[i] for i in range(len(numbers)-1, -1, -1) if
numbers[i] % 2 != 0]
```

```
print("Четные числа (в порядке возрастания индексов):", even_numbers)
print("Нечетные числа (в порядке убывания индексов):", odd_numbers)
```

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

```
def count_decreasing_subarrays(arr):
    n = len(arr)
    count = 0
    i = 0
    while i < n - 1:
        while i < n - 1 and arr[i] >= arr[i + 1]:
            i += 1
        if i < n - 1:
            count += 1
            i += 1
    return count
```

```
arr = [6, 5, 4, 8, 9, 7, 3, 2]
print(count_decreasing_subarrays(arr))
```

3. #Дано множество A из N точек на плоскости и точка B (точки заданы  
своими координатами x,y).

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

#Расстояние R между точками с координатами (x1, y1) и (x2, y2)

вычисляется по формуле:

# $R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

```
import math
```

```
def distance(x1, y1, x2, y2):
    return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
```

```
def closest_point(points, x, y):
    min_distance = float('inf')
    closest = None
    for point in points:
        dist = distance(point[0], point[1], x, y)
        if dist < min_distance:
            min_distance = dist
            closest = point
    return closest
```

### Протокол программы:

1. [422, 483, 279, 713, 506, 80, 818, 718, 635, 219, 801]

Четные числа (в порядке возрастания индексов): [422, 506, 80, 818, 718]

Нечетные числа (в порядке убывания индексов): [801, 219, 635, 713, 279, 483]

Process finished with exit code 0

2. 2

Process finished with exit code 0

3. Ближайшая точка из множества A: (3, 4)

Process finished with exit code 0

**Вывод:** В процессе выполнения практического занятия я выработал навыки составления программ линейной структуры в IDE PyCharm Community. Были использованы языковые конструкции. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub