

Студент группы Ис-27 Артебякин. К. Э

## Практическая работа № 6

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

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

**Постановка задачи:** Сформировать и вывести целочисленный список размера 10, содержащий степени двойки от первой до 10-й: 2, 4, 8, 16, ... .

**Тип алгоритма:** Циклический

**Текст программы:**

```
## Сформировать и вывести целочисленный список размера 10, содержащий степени
## двойки от первой до 10-й: 2, 4, 8, 16, ... .

# Генерируем значения от 2^1 до 2^10 с помощью списка, используя генератор списков.
powers_of_two = [2**i for i in range(1, 11)]

print(powers_of_two)
```

**Протокол работы программы:**

[2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]

Process finished with exit code 0

**Постановка задачи:** Дан список размера N. Найти номера тех элементов списка, которые больше своего левого соседа, и количество таких элементов. Найденные номера выводить в порядке их убывания.

**Тип алгоритма:** Циклический

**Текст программы:**

```
## Дан список размера N. Найти номера тех элементов списка, которые больше своего
## левого соседа, и количество таких элементов. Найденные номера выводить в
## порядке их убывания.

def find_greater_elements(lst): 1 usage  NiceKirill

    # Проверка на длину списка
    if len(lst) < 2:
        raise ValueError("Список должен содержать хотя бы два элемента.")

    # Список для хранения индексов элементов, которые больше своих левых соседей
    greater_indices = []

    for i in range(1, len(lst)):
        if lst[i] > lst[i - 1]:
            greater_indices.append(i)

    # Сортируем найденные индексы в порядке убывания
    greater_indices.sort(reverse=True)

    return greater_indices, len(greater_indices)

# Пример использования функции
lst = [3, 2, 7, 5, 6, 4]
indices, count = find_greater_elements(lst)
print("Индексы:", indices)
print("Количество:", count)
```

**Протокол работы программы:**

Индексы: [4, 2]

Количество: 2

Process finished with exit code 0

**Постановка задачи:** Дан список размера N. Обнулить элементы списка, расположенные между его минимальным и максимальным элементами (не включая минимальный и максимальный элементы).

**Тип алгоритма:** Циклический

**Текст программы:**

```
## Дан список размера N. Обнулить элементы списка, расположенные между его
## минимальным и максимальным элементами (не включая минимальный и
## максимальный элементы).

def zero_elements_between_min_max(lst): 1 usage  NiceKirill

    # Проверяем, что список содержит более одного элемента
    if len(lst) <= 1:
        raise ValueError("Список должен содержать более одного элемента")

    # Найдём индексы минимального и максимального элементов
    min_index = lst.index(min(lst))
    max_index = lst.index(max(lst))

    # Определим границы диапазона для обнуления
    start = min(min_index, max_index) + 1
    end = max(min_index, max_index)

    # Обнулим элементы между минимальным и максимальным значениями
    for i in range(start, end):
        lst[i] = 0

# Пример использования
lst = [3, 8, 9, 12, 15, 10, 11, 14, 13, 16, 17]
zero_elements_between_min_max(lst)
print(lst)
```

**Протокол работы программы:**

[3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 17]

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ линейной структуры в IDE PC. Были использованы языковые конструкции def, for, in, if, raise, return, reverse. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.