

## Практическое занятие № 6

**Тема:** составление программ с функциями 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 6.1

Даны целые числа  $N (>2)$ ,  $A$  и  $B$ . Сформировать и вывести целочисленный список размера 10, первый элемент которого равен  $A$ , второй равен  $B$ , а каждый последующий элемент равен сумме всех предыдущих.

**Тип алгоритма:** линейный.

#### Текст программы:

```
# Даны целые числа N (>2), A и B. Сформировать и вывести целочисленный список
# размера 10, первый элемент которого равен A, второй равен B, а каждый
# последующий элемент равен сумме всех предыдущих.
def generate_list (A, B):
    result = [A, B]
    for i in range(10):
        N = sum(result)
        result.append(N)
    return result
A=3
B=5
output_list = generate_list (A, B)
print(output_list)
```

#### Протокол работы программы:

[3, 5, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]

### Постановка задачи 6.2

Дан список размера  $N$ . Найти максимальный из его локальных минимумов (локальный минимум — это элемент, который меньше любого из своих соседей).

**Тип алгоритма:** линейный

#### Текст программы:

```
# Дан список размера N. Найти максимальный из его локальных минимумов
# (локальный минимум — это элемент, который меньше любого из своих соседей).
import random
def local_minima(arr):
    minima = []
    for i in range (1, len (arr) -1):
        if arr [i] < arr [i-1] and arr [i] < arr [i+1]:
            minima.append(arr[i])
    if len (minima) == 0:
        return "В списке нет локальных минимумов"
    else:
        return max(minima)
numbers = [random.randint(1, 100) for _ in range(10)]
print ("Исходный список: ", numbers)
result = local_minima(numbers)
```

```
print ("Максимальный из локальных минимумов: ", result)
```

### **Протокол работы программы:**

Исходный список: [16, 83, 11, 58, 35, 70, 98, 70, 66, 55]

Максимальный из локальных минимумов: 35

### **Постановка задачи 6.3**

список вправо на  $K$  позиций (при этом  $A_1$  перейдет в  $AK+1$ ,  $A_2$  — в  $AK+2$ , .. $AN-K$  — в  $AN$ , а исходное значение  $K$  последних элементов будет потеряно). Первые  $K$  элементов полученного списка положить равными 0.

**Тип алгоритма:** линейный

### **Текст программы:**

```
# список вправо на K позиций (при этом A1 перейдет в AK+1, A2 — в AK+2, ..AN-K — в
# AN, а исходное значение K последних элементов будет потеряно). Первые K
# элементов полученного списка положить равными 0.
```

```
def shift_elements(arr, k):
    n = len(arr)
    k = k%n
    shifted_arr =[0]*k
    for i in range(n-k):
        shifted_arr.append (arr[i])
    return shifted_arr
input_list =[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
k=3
result =shift_elements(input_list, k)
print("Результат: ", result)
```

### **Протокол работы программы:**

Результат: [0, 0, 0, 1, 2, 3, 4, 5, 6, 7]

**Вывод:** закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community и выложила на GitHub.