

## Практическая работа №6

**Тема:** составление программ со списками в IDE PyCharm Community

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с функциями в IDE PyCharm Community.

**Тип алгоритма:** Циклический

**Постановка задачи:**

### Задача 1.

Дан целочисленный список размера  $N$ . Увеличить все четные числа, содержащиеся в списке, на исходное значение первого четного числа. Если четные числа в списке отсутствуют, то оставить список без изменений

### Задача 2.

Дан список  $A$  размера  $N$ . Сформировать новый список  $B$  того же размера по следующему правилу: элемент  $B_K$  равен сумме элементов списка  $A$  с номерами от 1 до  $K$ .

### Задача 3.

Дан список размера  $N$  и целое число  $K$  ( $1 < K < N$ ). Осуществить сдвиг элементов списка вправо на  $K$  позиций (при этом  $A_1$  перейдет в  $A_{K+1}$ ,  $A_2$  — в  $A_{K+2}$ , ..  $A_{N-K}$  — в  $A_N$ , а исходное значение  $K$  последних элементов будет потеряно). Первые  $K$  элементов полученного списка положить равными 0.

**Текст программы:**

### Задача 1.

```
#Дан целочисленный список размера N. Увеличить все четные числа, содержащиеся  
#в списке, на исходное значение первого четного числа. Если четные числа в списке  
#отсутствуют, то оставить список без изменений.
```

```
def increase_even_numbers(lst):  
    # Создаем список четных чисел из исходного списка  
    even_numbers = [x for x in lst if x % 2 == 0]  
  
    # Если нет четных чисел, возвращаем исходный список без изменений  
    if not even_numbers:  
        return lst  
  
    # Берем первое четное число из списка четных чисел  
    first_even = even_numbers[0]  
  
    # Увеличиваем все четные числа на значение первого четного числа  
    result = [x + first_even if x % 2 == 0 else x for x in lst]
```

```

# Возвращаем результат
return result

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
result = increase_even_numbers(numbers)
print(result)

```

## Задача 2.

```

#Дан список A размера N. Сформировать новый список B того же размера по
#следующему правилу: элемент BK равен сумме элементов списка A с номерами от 1 до K.

A = [1, 2, 3, 4, 5] # Пример списка A
B = [] # Создаем пустой список B

sum_so_far = 0 # Переменная для хранения суммы элементов

for i in range(len(A)):
    sum_so_far += A[i] # Добавляем текущий элемент A[i] к сумме
    B.append(sum_so_far) # Добавляем сумму в список B

print(B) # Выводим список B

```

## Задача 3.

```

#Дан список размера N и целое число K (1 < K < N). Осуществить сдвиг элементов
#списка 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.

A = [1, 2, 3, 4, 5] # Пример списка A
N = len(A) # Размер списка A
K = 2 # Число K

B = [0] * K + A[:N-K] # Первые K элементов списка B равны 0, остальные элементы взяты
из списка A с учетом сдвига
print(B) # Вывод списка B

```

## Протокол работы программы:

### Задача 1.

```
[1, 4, 3, 6, 5, 8, 7, 10, 9, 12]
```

```
Process finished with exit code 0
```

### Задача 2.

```
[1, 3, 6, 10, 15]
```

```
Process finished with exit code 0
```

### Задача 3.

```
[0, 0, 1, 2, 3]
```

```
Process finished with exit code 0
```

**Вывод:** в процессе выполнения практического занятия выработал(а) навыки составления программ циклической структуры в IDE PyCharm Community. Были использованы языковые конструкции `def`, `for`, `if`, `return`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.