

## Практическое занятие № 6

**Тема:** 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 список A размера N. Вывести его элементы в следующем порядке: A1, AN, A2, AN-1, A3, AN-2, ....

**Тип алгоритма:** циклический.

### Текст программы:

```
# Дан список A размера N. Вывести его элементы в следующем порядке: A1, AN, A2,
# AN-1, A3, AN-2, ...

N = int(input("Введите размер списка: "))
A = []
for _ in range(N):
    A.append(int(input("Введите элемент списка: ")))

for i in range(N // 2):
    print(A[i], A[N - 1 - i], end=' ')
if N % 2 != 0:
    print(A[N // 2])
```

### Протокол программы:

Введите размер списка: 5

Введите элемент списка: 1

Введите элемент списка: 2

Введите элемент списка: 3

Введите элемент списка: 4

Введите элемент списка: 5

1 5 2 4 3

**Постановка задачи:**

Дан целочисленный список размера N, все элементы которого упорядочены (по возрастанию или по убыванию). Найти количество различных элементов в данном списке.

**Тип алгоритма:** циклический.

**Текст программы:**

```
# Дан целочисленный список размера N, все элементы которого упорядочены (по
# возрастанию или по убыванию). Найти количество различных элементов в данном
# списке.

N = int(input("Введите размер списка: "))
A = []

for _ in range(N):
    while True:
        try:
            element = int(input("Введите элемент списка: "))
            A.append(element)
            break
        except ValueError:
            print("Ошибка: Введите целое число.")

A.sort()

unique_count = 1
for i in range(1, N):
    if A[i] != A[i - 1]:
        unique_count += 1

print("Количество различных элементов:", unique_count)
```

**Протокол программы:**

Введите размер списка: 5

Введите элемент списка: 11

Введите элемент списка: 11

Введите элемент списка: 22

Введите элемент списка: 33

Введите элемент списка: 11

Количество различных элементов: 3

**Постановка задачи:**

Дан список размера N. Осуществить циклический сдвиг элементов списка влево на одну позицию (при этом  $A_N$  перейдет в  $A_{N-1}$ ,  $A_{N-1}$  — в  $A_{N-2}$ , . . .,  $A_1$  — в  $A_N$ ).

**Тип алгоритма:** циклический.

**Текст программы:**

```
# Дан список размера N. Осуществить циклический сдвиг элементов списка влево на
# одну позицию (при этом  $A_N$  перейдет в  $A_{N-1}$ ,  $A_{N-1}$  — в  $A_{N-2}$ , . . .,  $A_1$  — в  $A_N$ ).

N = int(input("Введите размер списка: "))
A = []
for _ in range(N):
    A.append(int(input("Введите элемент списка: ")))

first_element = A[0]
for i in range(N - 1):
    A[i] = A[i + 1]
A[N - 1] = first_element

print("Список после циклического сдвига:", A)
```

**Протокол программы:**

Введите размер списка: 5

Введите элемент списка: 1

Введите элемент списка: 2

Введите элемент списка: 3

Введите элемент списка: 4

Введите элемент списка: 5

Список после циклического сдвига: [2, 3, 4, 5, 1]

**Вывод:**

В процессе выполнения практического занятия я выработал навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции for, while, if, try-except, а также методы списков и операции с числами. Выполнены разработка кода, отладка, тестирование и оптимизация программного кода.

Программа успешно выводит элементы списка в определенном порядке, как показано в примере вывода. Для этого был введен размер списка и его элементы, и программа корректно выводила элементы в порядке  $A_1, A_N, A_2, A_{N-1}, A_3, A_{N-2}$ , и так далее.

Также были разработаны программы, которые определяли количество различных элементов в упорядоченном целочисленном списке и осуществляли циклический сдвиг элементов списка влево на одну позицию. Программы успешно выводили результаты после ввода размера списка и его элементов.

Готовые программные коды выложены на GitHub.