

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: дан список A размера N. Вывести его элементы в следующем порядке: A1, A2, AN, AN-1, A3, A4, AN-2, AN-3,

Тип алгоритма: Циклический

Текст программы:

```
def print_list(A):
    if not isinstance(A, list) or len(A) == 0: # проверка на корректность
        входных данных (не явл ли список пустым)
        raise ValueError("Список не должен быть пустым!")

    n = len(A) #длина списка
    b = [] #новый список

    i = 0 #первый индекс
    j = n - 1 #последний индекс

    while i <= j: # пока i меньше или равен j
        if i != j: # если различны
            b.append(A[i]) #добавляем первый эл
            b.append(A[j]) #добавляем последний эл

            # если длина списка нечетная, добавляем центральный элемент только
            один раз
            elif i == j:
                b.append(A[i])

            i += 1
            j -= 1

    return b #возвращение нового списка

A = [1, 2, 3, 4, 5]
spisok = print_list(A)
print(spisok)
```

Протокол работы программы:

[1, 5, 2, 4, 3]

Process finished with exit code 0

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

Текст программы:

```
def count_unique_elements(A):
    if not isinstance(A, list) or len(A) == 0: # проверка на корректность
        входных данных (не явл ли список пустым)
        raise ValueError("Список не должен быть пустым!")

    return len(set(A)) #удаление повторных элементов + кол оставшихся
    элементов

A = [1, 1, 2, 2, 3, 4, 4, 5]
B = count_unique_elements(A)
print("Количество уникальных элементов: ", B)
```

Протокол работы программы:

Количество уникальных элементов: 5

Process finished with exit code 0

Постановка задачи: дан список A размера N и целое число K ($1 < K < 4$, $K < N$). Осуществить циклический сдвиг элементов списка вправо на K позиций (при этом A₁ перейдет в A_{K+1}, A₂ — в A_{K+2}, ..., A_N — в A_K). Допускается использовать вспомогательный список из 4 элементов

Текст программы:

```
def cyclic_shift_right(A, K): #цикл сдвиг вправо
    N = len(A)

    if not (1 < K < 4 and K < N): #проверка условия K
        raise ValueError("Значение K должно удовлетворять условиям!")

    time = A[:K] # временный список + срез от 0 до K-1

    for i in range(N - K):
        A[i] = A[i + K] # сдвиг остальных элементов вправо на K позиций

    for i in range(K):
        A[N - K + i] = time[i] # заполнение последних K позиций элементами из
    врем списка

A = [1, 2, 3, 4]
K = 3
cyclic_shift_right(A, K)
print(A)
```

Протокол работы программы:

[4, 1, 2, 3]

Process finished with exit code 0