

Практическое задание № 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 №1.

Дан список размера N и целые числа K и L ($1 < K < L < N$).

Найти среднее арифметическое элементов списка с номерами от K до L включительно.

Постановка задачи №2.

Дан целочисленный список размера N , содержащий ровно два одинаковых элемента.

Найти номера одинаковых элементов и вывести эти номера в порядке возрастания.

Постановка задачи №3.

Дан список размера N , все элементы которого, кроме первого, упорядочены по возрастанию. Сделать список упорядоченным, переместив первый элемент на новую позицию.

Студент группы ИС-27 Орешников И.Д.

Текст программы №1:

```
#Дан список размера N и целые числа K и L ( $1 < K < L < N$ ). Найти среднее
#арифметическое элементов списка с номерами от K до L включительно.

def average_of_sublist(A, K, L):
    sublist = A[K-1:L] # Получаем подсписок с элементами от K до L
    average = sum(sublist) / len(sublist) # Считаем среднее значение подсписка
    return average

A = [3, 1, 5, 2, 7, 5]
K = 2
L = 3

result = average_of_sublist(A, K, L)
print(result)
```

Протокол работы №1:

3.0

Process finished with exit code 0

Текст программы №2:

```
#Дан целочисленный
#список размера N, содержащий ровно два одинаковых элемента.
#Найти номера одинаковых элементов и вывести
#эти номера в порядке возрастания.

def find_duplicate_indices(nums):
    dd = {}
    for i, num in enumerate(nums):
        if num in dd:
            return (dd[num], i)
        else:
            dd[num] = i

my_list = [3, 1, 5, 2, 7, 5]
result = find_duplicate_indices(my_list)
print(result)

print(f"Длина незанятой части отрезка A равна {result}")
else:
    print("Пожалуйста, введите положительные числа, где A > B.")
```

Протокол работы №2:

(2, 5)

Process finished with exit code 0

Текст программы №2:

```
#Дан список размера N, все элементы которого, кроме первого,
#не являются нулями. Упорядочить список упорядоченным, переместив первый
#элемент на новую позицию.
def order_list(lst):
    # Проверка наличия элементов в списке
    if len(lst) > 1:
        # Перемещение первого элемента на новую позицию
        first_element = lst.pop(0)
        lst.insert(1, first_element) # Сортировка списка
        lst.sort()
    return lst

my_list = [3, 2, 4, 6, 8]

result = order_list(my_list)
print("Упорядоченный список:", result)
```

Протокол работы №3:

Упорядоченный список: [2, 3, 4, 6, 8]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ со списками в IDE PyCharm Community. Выполнены разработка кода , отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.