

Практическое задание №6

Тема: составление программ ветвящейся структуры в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры 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. 2) Дан целочисленный список размера N. Найти максимальное количество его одинаковых элементов. 3) Дан список размера N, все элементы которого, кроме одного, упорядочены по убыванию. Сделать список упорядоченным, переместив элемент, нарушающий упорядоченность, на новую позицию.

Тип алгоритма: циклический

Текст программы:

1)

```
def average_between_k_and_l(lst, k, l):
    # Проверка на корректность входных данных
    if not 1 < k < l < len(lst):
        return "Некорректные значения K и L"

    # Выбор подсписка с элементами от K до L
    sublist = lst[k-1:l]

    # Вычисление среднего арифметического
    average = sum(sublist) / len(sublist)

    return average

# Пример использования
N = int(input("Введите размер списка N: "))
my_list = [int(input(f"Введите элемент {i+1}: ")) for i in range(N)]
K = int(input("Введите значение K (1 < K < L < N): "))
L = int(input("Введите значение L (1 < K < L < N): "))

result = average_between_k_and_l(my_list, K, L)

if isinstance(result, str):
    print(result)
else:
    print(f"Среднее арифметическое элементов с номерами от {K} до {L}: {result}")
```

2)

```
def max_occurrences(lst):  
    if not lst:  
        return 0  
  
    occurrences = {}  
    max_count = 0  
  
    for num in lst:  
        occurrences[num] = occurrences.get(num, 0) + 1  
        max_count = max(max_count, occurrences[num])  
  
    return max_count  
  
# Пример использования  
N = int(input("Введите размер списка N: "))  
my_list = [int(input(f"Введите элемент {i+1}: ")) for i in range(N)]  
  
result = max_occurrences(my_list)  
print(f"Максимальное количество одинаковых элементов в списке:  
{result}")
```

3)

```
testing_list = []
```

```

list_range = 4

for i in range(list_range):
    testing_list.append(int(input(f"list index {i+1}: ")))

previous = None

for i in testing_list:
    if previous:
        if previous <= i:
            testing_list.remove(previous)
            testing_list = [previous,*testing_list]
        previous = i

```

Протокол работы программы:

1) /usr/bin/python3.9 /home/student/Документы/pzuc17Fuze/pz 6/pz 6.1.py

Введите размер списка N: 3

Введите элемент 1: 2

Введите элемент 2: 1

Введите элемент 3: 3

Введите значение K ($1 < K < L < N$): 5

Введите значение L ($1 < K < L < N$): 3

Некорректные значения K и L

Process finished with exit code 0

2) /usr/bin/python3.9 /home/student/Документы/pzuc17Fuze/pz 6/pz 6.2.py

Введите размер списка N: 3

Введите элемент 1: 2

Введите элемент 2: 1

Введите элемент 3: 3

Максимальное количество одинаковых элементов в списке: 1

Process finished with exit code 0

3)

```
/usr/bin/python3.9 /home/student/Документы/pzuc17Fuze/pz 6/pz 6.3.py
```

```
list index 1: 3
list index 2: 2
list index 3: 2
list index 4: 1
[2, 3, 2, 1]

Process finished with exit code 0
```

Вывод:

В процессе выполнения практического задания выработал навыки составления программ циклической структуры в IDE PyCharm Community. Выполнены разработки кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.