

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..

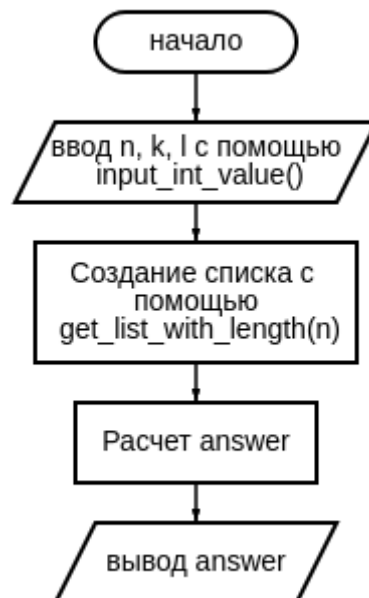
Задание 1.

Постановка задачи.

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

Тип алгоритма: линейный.

Блок схема алгоритма:



Текст программы:

```
import random

def input_int_value(argument: str):
```

```

# Int check
while type(incoming_value := input(f"Input your {argument}-number: ")) != int:
    try:
        incoming_value = int(incoming_value)
    except ValueError:
        print("! Your input must be int-type.")
    else:
        return incoming_value

def get_list_with_length(n: int):
    return [random.randint(0, 1) for _ in range(n)]

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

n_value = input_int_value("N")
k_value = input_int_value("K")
l_value = input_int_value("L")
a = get_list_with_length(n_value)
answer = sum(a[k_value:l_value + 1])
print(f"The amount of {a} from {k_value} to {l_value} indexes is: {answer}.")

```

Протокол работы программы:

```

Input your N-number: 12
Input your K-number: 2
Input your L-number: 9
The amount of [0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0] from 2 to 9 indexes is: 4.

```

Задание 2.

Постановка задачи.

Дан целочисленный список размера N. Найти количество различных элементов в данном списке.

Тип алгоритма: линейный.

Блок схема алгоритма:



Текст программы:

```
import random

def input_int_value(argument: str):
    # Int check
    while type(incoming_value := input(f"Input your {argument}-number: ")) != int:
        try:
            incoming_value = int(incoming_value)
        except ValueError:
            print("(!) Your input must be int-type.")
        else:
            return incoming_value

def get_list_with_length(n: int):
    return [random.randint(-10, 10) for _ in range(n)]

# 2. Дан целочисленный список размера N.
# Найти количество различных элементов в данном списке.

n_value = input_int_value("N")
a = get_list_with_length(n_value)
unique_items_number = len(set(a))
print(f"The number of unique items in {a} is: {unique_items_number}.")
```

Протокол работы программы:

```
Input your N-number: 14
The number of unique items in [-10, 6, -2, 8, 1, 5, 8, -1, 2, -9, -3, -3, 6, 5] is: 10.
```

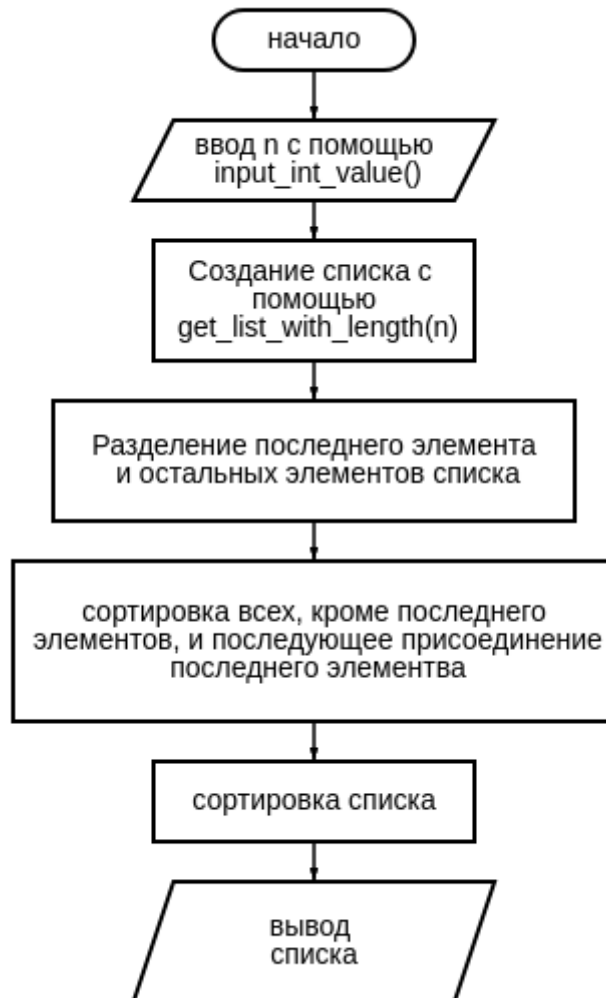
Задание 3.

Постановка задачи.

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

Тип алгоритма: линейный.

Блок схема алгоритма:



Текст программы:

```
import random

def input_int_value(argument: str):
    # Int check
    while type(incoming_value := input(f"Input your {argument}-number: ")) != int:
        try:
            incoming_value = int(incoming_value)
        except ValueError:
            print("! Your input must be int-type.")
        else:
            return incoming_value

def get_list_with_length(n: int):
    return [random.randint(-10, 10) for _ in range(n)]

# 3. Дан список размера N, все элементы которого, кроме последнего, упорядочены по
# возрастанию. Сделать список упорядоченным, переместив последний элемент на
# новую позицию.

n_value = input_int_value("N")
*a, b = get_list_with_length(n_value)
c = sorted(a) + [b]
print(f"Sorted {c} is {sorted(c)}")
```

Протокол работы программы:

```
Input your N-number: 10
Sorted [-9, -8, -3, -3, 3, 3, 6, 9, 10, 2] is [-9, -8, -3, -3, 2, 3, 3, 6, 9, 10].
```

Вывод: в процессе выполнения практического занятия я выработал навыки составления программ *со списками* в *IDE PyCharm Community*. Были использованы языковые конструкции: *while*, *for*, *try*.
Выполнены разработка, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на *GitHub*.