

Практическое занятие №6

Тема: Составление программ с функциями 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 целочисленный список A размера 10. Вывести порядковый номер последнего из тех его элементов $A(k)$, которые удовлетворяют двойному неравенству $A(1) < A(k) < A(10)$. Если таких элементов нет, то вывести 0.
2. Дано число R и список размера N. Найти два различных элемента списка, сумма которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания их индексов (определение наиболее близких чисел - то есть такой элемент $A(k)$, для которого величина $|A(k) - R|$ является минимальной).
3. Дан список размера N, все элементы которого, кроме первого, упорядочены по возрастанию. Сделать список упорядоченным, переместив первый элемент на новую позицию.

Текст программы:

Задание 1

#дан целочисленный список A размера 10. Вывести порядковый номер последнего из тех его элементов A_k , которые удовлетворяют двойному неравенству $A_1 < A_k < A_{10}$. Если таких элементов нет, то вывести 0.

```
def find_last_element(A): #нахождение последнего элемента
    try:
        last_element = A[-1]
        if 1 < last_element < 10:
            return last_element
        else:
            return 0
    except IndexError:
        return 0

A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
result = find_last_element(A)
print("Последний элемент, удовлетворяющий условию:", result)
```

Задание 2

#дано число R и список размера N. Найти два различных элемента списка, сумма которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания их индексов (определение наиболее близких чисел - то есть такой элемент $A(k)$, для которого величина $|A(k) - R|$ является минимальной)

```
import numpy as np

def find_closest_elements(lst, R): #функция принимает список lst и значение R
    arr = np.array(lst) #преобразование списка в массив NumPy
    min_pos = np.argmin(arr) #поиск индексов минимального и максимального элементов
    max_pos = np.argmax(arr)

    if np.abs(arr[min_pos] - R) < np.abs(arr[max_pos] - R): #сравнение расстояний от заданного значения R до
        минимального и максимального элементов
        return lst[min_pos], lst[min_pos + 1] #возвращение самых близких элементов к заданному значению
    else:
        return lst[max_pos], lst[max_pos - 1]

#пример использования
lst = [1, 2, 3, 4, 5]
R = 3
print(find_closest_elements(lst, R))
```

Протокол работы программы:

Задание 1

Последний элемент, удовлетворяющий условию: 0
Process finished with exit code 0

Задание 2

(5, 4)
Process finished with exit code 0

Задание 3

[3, 1, 2, 4, 5, 6, 7, 8, 9, 10]

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community.
Выполнены разработка кода, отладка, тестирование.