

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 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.

Тип алгоритма: циклический

Текст программы:

```
A = [2, 12, 3, 12, 99, 1, 5, 86, 15, 33]
itFinds = False
B = []
for i in range(0, 10):
    if A[0] < A[i] < A[9]:
        itFinds = True
        B.append(i)

print(B[-1])

if not itFinds:
    print(0)
```

Протокол работы программы:

8

Process finished with exit code 0

Задание 2:

Постановка задачи.

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).

Тип алгоритма: циклический

Текст программы:

```
R = 62
A = [1, 12, 43, 65, 18, 35, 44]

def find_closest_sum_pair(R, numbers):
    closest_sum = float() # Инициализируем переменную для
    хранения наименьшей разницы между суммой и R
    result_pair = [None, None] # Переменная для хранения
    найденной пары

    for i in range(len(numbers)):
        for j in range(i+1, len(numbers)):
            current_sum = numbers[i] + numbers[j]
            if abs(current_sum - R) < abs(closest_sum - R):
                closest_sum = current_sum
                result_pair = [i, j]
    return result_pair

result = find_closest_sum_pair(R, A)
```

Протокол работы программы:

Две ближайшие к числу 62 суммирующиеся числа (их индексы): [4, 6]

Process finished with exit code 0

Задание 3:

Постановка задачи.

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

Тип алгоритма: циклический

Текст программы:

```
A = [11, 4, 14, 23, 45, 56, 101, 200]
A.sort()
print(A)
```

Протокол работы программы:

[4, 11, 14, 23, 45, 56, 101, 200]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ со списками в IDE PyCharm Community.