

Практическое занятие №6

Тема: Наименование практического занятия: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ со списками в IDE PyCharm Community.

Постановка задачи 1: Дан целочисленный список размера 10. Вывести все содержащиеся в данном списке четные числа в порядке убывания их индексов, а также их количество K.

Тип алгоритма: ветвление

Дан целочисленный список размера 10. Вывести все содержащиеся в данном списке
четные числа в порядке убывания их индексов, а также их количество K.

```
try:
    a = [1, 5, 2, 3, 4, 8, 9, 7, 10]
    b = []
    for i in range(len(a)-1, 0, -1):
        if a[i] % 2 == 0:
            b.append(a[i])
    k = len(b)
    print("Количество четных чисел в порядке убывания: ", b)
    print("Количество чисел:", k)
except ValueError:
    print('Ошибка')
```

Протокол работы программы:

Количество четных чисел в порядке убывания: [10, 8, 4, 2]

Количество чисел: 4

Process finished with exit code 0

Постановка задачи 2: Дан список размера N. Найти количество участков, на которых его элементы монотонно возрастают.

Тип алгоритма: ветвление

Дан список размера N. Найти количество участков, на которых его элементы
монотонно возрастают.

```
try:
    n = [-1, -3, 4, -10, 3, -2, 4, -45]

    def vsrcst(n):
        v = 0
        s = 1
        for i in range(1, len(n)):
            if n[i] > n[i - 1]:
                s += 1
            else:
                if s > 1:
                    v += 1
                    s = 1
        if s > 1:
            v += 1
        return v
    vsrcst(n)
    print("Количество участков элементы которых возрастают: ", vsrcst(n))
except ValueError:
    print("Ошибка")
```

Протокол работы программы:

Количество участков элементы которых возрастают: 3

Process finished with exit code 0

Студент группы ИС-28 Данилов Д.М.

Постановка задачи 3:

Тип алгоритма: циклический

```
# Дан список размера N. Заменить каждый элемент списка на среднее арифметическое  
# этого элемента и его соседей.
```

```
try:  
    s = [-1, 0.7, 84, 2.5, 396]  
  
    def srd_arfm(n):  
        d = len(n)  
        b = []  
        for i in range(d):  
            if i == 0:  
                v = (n[i] + n[i + 1]) / 2  
            elif i == d - 1:  
                v = (n[i - 1] + n[i]) / 2  
            else:  
                v = (n[i - 1] + n[i] + n[i + 1]) / 3  
            b.append(v)  
        return b  
    g = srd_arfm(s)  
    print(g)  
except ValueError:  
    print("Ошибка")
```

Протокол работы программы:

```
[-0.15000000000000002, 27.900000000000002, 29.066666666666666,  
160.83333333333334, 199.25]
```

Process finished with exit code 0

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