

Практическое задание № 6

Тема: Составление программ со списками в IDE PyCharm Community .

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

Постановка задачи №1.

Сформировать и вывести целочисленный список размера 10, содержащий 10 первых положительных нечетных чисел: 1,3,5,...

Тип алгоритма: циклический.

Текст программы:

```
# Сформировать и вывести целочисленный список размера 10, содержащий 10
# первых положительных нечетных чисел: 1,3,5,...
a = []
i = 0
for i in range(10):
    i += 1*i+1
    a.append(i)
print(a)
```

Протокол работы программы:

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

Process finished with exit code 0

Постановка задачи №2.

Дан список размера N. Найти номера тех элементов, которые больше своего правого соседа, а также количество таких элементов. Найденные номера выводить в порядке их возрастания.

Тип алгоритма: циклический.

Текст программы:

```
# Дан список размера N. Найти номера тех элементов, которые больше своего
# правого соседа, а также количество таких элементов.
# Найденные номера выводить в порядке их возрастания.
import random
list = []
result = []
i = 0
try:
    N = int(input("Введите размер списка: "))
    if N > 0:
        while i < N:
            list.append(random.randint(0,100))
            i += 1
        for i in range(len(list) - 1):
            if list[i] > list[i + 1]:
                result.append(i)
        print("Список: ",list)
        print("Индексы элементов, которые больше своего правого соседа:",result)
        print("Количество элементов:", len(result))
    else: print('Необходимо ввести положительное число')
except ValueError :
    print("Неправильный тип данных")
```

Протокол работы программы:

Введите размер списка: 12

Список: [58, 10, 38, 67, 43, 47, 30, 94, 76, 11, 99, 77]

Индексы элементов, которые больше своего правого соседа: [0, 3, 5, 7, 8, 10]

Количество элементов: 6

Process finished with exit code 0

Постановка задачи №3.

Дан список A размера N и целые числа K и L ($1 < K < L < N$). Переставить в обратном порядке элементы списка, расположенные между элементами A_K и A_L , не включая эти элементы.

Тип алгоритма: циклический.

Текст программы:

```
# Дан список A размера N и целые числа K и L (1 < K < L < N). Переставить в  
# обратном порядке элементы списка, расположенные между элементами A_K и A_L,  
# не включая эти элементы.  
import random  
i = 0  
A = []  
try:  
    N = int(input("Введите размер списка: "))  
    if N > 0:  
        while i < N:  
            A.append(random.randint(0,100))  
            i += 1  
        print('Список:',A)  
    else:print()  
    K = int(input('Введите K: '))  
    L = int(input('Введите L: '))  
    if 1 < K < L < N:  
        sublist = A[K+1:L] # подсписок между индексами K и L  
        reversed_sublist = reversed(sublist) # перевернутый подсписок  
        A[K+1:L] = reversed_sublist  
        print("Измененный список: ",A)  
    else: print('Данные введены неверно(1 < K < L < N)')  
except ValueError:  
    print("Неправильный тип данных")
```

Протокол работы программы:

Введите размер списка: 12

Список: [40, 14, 13, 0, 99, 23, 23, 78, 60, 9, 28, 16]

Введите K: 2

Введите L: 6

Измененный список: [40, 14, 13, 23, 99, 0, 23, 78, 60, 9, 28, 16]

Process finished with exit code 0

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

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовый программный код выложен на GitHub.