

Практическое занятие №6

Тема: составление программ со списками в IDE PyCharmCommunity.

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

Постановка задачи №1: Дан список A размера N (N — нечетное число). Вывести его элементы с нечетными номерами в порядке убывания номеров: $A_N, A_{N-2}, A_{N-4}, \dots, A_1$. Условный оператор не использовать.

Тип алгоритма: циклический

Текст программы:

```
# Дан список A размера N (N — нечетное число). Вывести его элементы с
# нечетными
# номерами в порядке убывания номеров:  $A_N, A_{N-2}, A_{N-4}, \dots, A_1$ . Условный
# оператор не
# использовать.
from random import randint
N = int(input('Введите длину списка (нечётное число): '))
b = []
for i in range(N): # берём случайные числа от 1 до 100
    a = randint(1, 100)
    b.append(a)
print(f'Список - {b}')
print('Элементы с нечётными номерами в порядке убывания номеров: ')
a = [b[j] for j in range(1, N, 2)] # перебираем числа
a.reverse() # переворачиваем
print(a)
```

Протокол работы программы:

Введите длину списка (нечётное число): 5

Список - [63, 11, 92, 30, 93]

Элементы с нечётными номерами в порядке убывания номеров:

[30, 11]

Process finished with exit code 0

Постановка задачи №2: Дан список размера N. Найти два соседних элемента, сумма которых максимальна, и вывести эти элементы в порядке возрастания их индексов.

Текст программы:

```
# Дан список размера N. Найти два соседних элемента, сумма которых
максимальна, и
# вывести эти элементы в порядке возрастания их индексов.
from random import randint
N = int(input('Сколько элементов в списке?: '))
b = [] # создаём пустой список
for i in range(N): # берём случайные числа от 1 до 100
    a = randint(1, 100)
    b.append(a)
print(b)
k = 1
max_sum = b[k] + b[k + 1]
for i in range(1, N):
    if b[i - 1] + b[i] > max_sum:
        max_sum = b[i - 1] + b[i]
        k = i - 1
print('Два соседних элемента сумма которых максимальна: 'f'{b[k]} {b[k+1]}')
```

Протокол работы программы:

Сколько элементов в списке?: 10

[58, 48, 47, 53, 90, 65, 45, 33, 40, 38]

Два соседних элемента сумма которых максимальна: 90 65

Process finished with exit code 0

Постановка задачи №3: Дан список размера N, все элементы которого, кроме последнего, упорядочены по возрастанию. Сделать список упорядоченным, переместив последний элемент на новую позицию.

Тип алгоритма: циклический

Текст программы:

1. Дан список размера N и целое число K ($1 < K < N$). Осуществить сдвиг элементов списка вправо на K позиций (при этом A_1 перейдет в A_{K+1} , A_2 — в A_{K+2} , .. A_{N-K} — в A_N , а исходное значение K последних элементов будет потеряно). Первые K элементов полученного списка положить равными 0.

```
2. # Дан список размера N и целое число K ( $1 < K < N$ ). Осуществить сдвиг
элементов списка
# вправо на K позиций (при этом  $A_1$  перейдет в  $A_{K+1}$ ,  $A_2$  — в  $A_{K+2}$ , ..  $A_{N-K}$ 
— в  $A_N$ , а исходное
# значение K последних элементов будет потеряно). Первые K элементов
полученного
# списка положить равными 0.
from random import randint
n = int(input('Введите длину списка: '))
mylist = [] # создаём пустой список
for i in range(n): # берём случайные числа от 1 до 100
    a = randint(1, 100)
    mylist.append(a)
print(mylist)
x = int(input('На сколько элементов осуществить сдвиг: '))
mylist = mylist[-x:] + mylist[:-x]
print(*mylist)
```

Протокол работы программы:

Введите длину списка: 10

[11, 32, 77, 37, 60, 63, 46, 93, 35, 77]

На сколько элементов осуществить сдвиг: 5

63 46 93 35 77 11 32 77 37 60

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. Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.