

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.

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.

Текст программы:

Вариант 12. 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.

```
while True:
```

```
    try: # обработка исключений
        n = int(input('Введите нечетное число: '))
        n += (n % 2 == 0) * 1 # если n четное, то прибавляем 1
        break
    except ValueError:
        print('Вы ввели не число!')
```

```
a = list(range(1, n + 1)) # список от 1 до N
result = a[::-2] # элементы с нечетными номерами в убывании
print(f'Исходный список: {a}')
print(f'Результат: {result}')
```

Протокол работы программы:

Введите нечетное число: 5

Исходный список: [1, 2, 3, 4, 5]

Результат: [5, 3, 1]

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:

Вариант 12. Дан список размера N. Найти два соседних элемента, сумма которых
максимальна,

и вывести эти элементы в порядке возрастания их индексов.

```
def pairSum(numbers):
    if len(numbers) < 2:
        return None

    maxSum = 0
    maxPair = None

    for i in range(len(numbers) - 1):
        tekushayaSum = numbers[i] + numbers[i + 1]
        if tekushayaSum > maxSum:
            maxSum = tekushayaSum
            maxPair = (numbers[i], numbers[i + 1])
    return maxPair

import random

N = input("Введите размер списка N: ")
while type(N) != int: # обработка исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите размер списка N: ")

a = [random.randint(1, 100) for i in range(N)] # список рандомных чисел

print("Сгенерированный список:", a)

result = pairSum(a)
if result:
    print("Пара с макс. суммой:", result)
else:
    print("Список слишком короткий")
```

Протокол работы программы:

Введите размер списка N: 5

Сгенерированный список: [89, 71, 63, 12, 16]

Пара с макс. суммой: (89, 71)

Process finished with exit code 0

Постановка задачи 3

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.

Текст программы:

*# Вариант 12. 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.*

```
N = input("Введите размер списка N: ")
while type(N) != int: # обработка исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите размер списка N: ")
```

```
K = input("Введите размер списка K: ")
while type(K) != int: # обработка исключений
    try:
        K = int(K)
        if not (1 < K < N):
            K = input("Введите размер списка K: ")
    except ValueError:
        K = input("Введите размер списка K: ")
```

```
A = list(range(1, N + 1)) # генерация списка от 1 до N
print("Исходный список:", A)
```

```
A = [0] * K + A[:-K] # сдвиг вправо на K
```

```
print("Список после сдвига:", A)
```

Протокол работы программы:

Введите размер списка N: 5

Введите размер списка K: 2

Исходный список: [1, 2, 3, 4, 5]

Список после сдвига: [0, 0, 1, 2, 3]

Process finished with exit code 0

Вывод: я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции while, try, except. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.