

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 список A размера N . Вывести вначале его элементы с четными номерами (в порядке возрастания номеров), а затем — элементы с нечетными номерами (также в порядке возрастания номеров): $A_2, A_4, A_6, \dots, A_1, A_3, A_5, \dots$. Условный оператор не использовать.

Тип алгоритма.

Линейный.

Текст программы.

```
"""
Дан список A размера N. Вывести вначале его элементы с четными номерами (в
порядке возрастания номеров), а затем — элементы с нечетными номерами (также в
порядке возрастания номеров): A2, A4, A6, . . . , A1, A3, A5, ... . Условный оператор не
использовать.
"""

while True:

    try:

        A = int(input("Введите число >>> "))

        list = []

        for i in range(1, A + 1):

            list.append(i)

        print("Четные числа: ", list[1::2])

        print("Нечетные числа: ", list[:2])

        break

    except ValueError:

        print("Вы ввели не число")
```

Протокол работы программы.

Введите число >>> 23

Четные числа: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]

Нечетные числа: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23]

Process finished with exit code 0

Задание 2.

Постановка задачи.

Дано число R и список размера N . Найти два соседних 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.

Линейный.

Текст программы

```
"""
Дано число R и список размера N. Найти два соседних 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 AK, для
которого величина |AK - R| является минимальной).
"""

import random

def find_closest_elements(R, arr):

    min_diff = float('inf')

    res = None

    for i in range(len(arr) - 1):

        diff = abs(arr[i] + arr[i+1] - R)

        if diff < min_diff:

            min_diff = diff

            res = (i, i+1)

    return res

R = int(input('Введите число >>> '))

arr = [random.randint(-10, 50) for _ in range(int(input('Введите количество элементов списка >>> ')))]

print(arr)

result = find_closest_elements(R, arr)

if result:

    print("Ближайшие соседние индексы списка >>> ", result[0], result[1])
```

Протокол работы программы.

Введите число >>> 12

Введите количество элементов списка >>> 8

[31, 23, 37, -4, 35, 15, 21, 15]

Ближайшие соседние индексы списка >>> 3 4

Process finished with exit code 0

Задание 3.

Постановка задачи.

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

Тип алгоритма.

Линейный.

Текст программы

```
"""
Дан список размера N и целое число K (1 < K < N). Осуществить сдвиг элементов
списка влево на K позиций (при этом AN перейдет в AN-K, AN-1 — в AN-K-1, ..AK+1 — в
A1, а исходное значение K первых элементов будет потеряно). Последние K
элементов полученного списка положить равными 0.
"""

def shift_and_zero(lst, k):

    new_lst = [0] * len(lst)

    for i in range(len(lst) - k):

        new_lst[i - k] = lst[i]

    return new_lst

lst = [1,3,5,6,7,4,9]

k = 2

if k < 1 or k >= len(lst):

    print('Ошибка! Введено неверное число k')

else:

    print(shift_and_zero(lst, k))
```

Протокол работы программы.

[5, 6, 7, 0, 0, 1, 3]

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составление программ со списками в IDE PyCharm Community.