

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. Вывести его элементы в следующем порядке: A1, A2, AN, AN-1, A3, A4, AN-2, AN-3,

Тип алгоритма: Циклический

Текст программы:

```
1  #Дан список A размера N. Вывести его элементы в следующем порядке: A1, A2, AN,
2  #AN-1, A3, A4, AN-2, AN-3, ... .
3
4
5  def possitive(A): 1usage new *
6      n = len(A)
7      result = []
8      left = 0
9      right = n - 1
10
11     while left <= right:
12         if left == right:
13             result.append(A[left])
14         else:
15             result.append(A[left])
16             result.append(A[right])
17
18         left += 1
19         right -= 1
20
21     return result
22
23 # Пример использования
24 try:
25     N = int(input("Введите размер списка N: "))
26     if N <= 0:
27         print("Ошибка: Размер списка должен быть положительным целым числом.")
28     else:
29         A = []
30
31         # Заполняем список A
32         for i in range(N):
33             value = int(input(f"Введите элемент A[{i + 1}]: "))
34             A.append(value)
35
36         rearranged = possitive(A)
37
38         # Выводим результат
39         print("Элементы списка в новом порядке:")
40         print(rearranged)
41 except ValueError:
42     print("Ошибка: Пожалуйста, введите целое число.")
```

Протокол программ:

Введите размер списка N: 6

Введите элемент A[1]: 1

Введите элемент A[2]: 2

Введите элемент A[3]: 3

Введите элемент A[4]: 4

Введите элемент A[5]: 5

Введите элемент A[6]: 6

Элементы списка в новом порядке:

[1, 6, 2, 5, 3, 4]

Process finished with exit code 0

Постановка задачи:

2. Дан целочисленный список размера N, содержащий ровно два одинаковых элемента. Найти номера одинаковых элементов и вывести эти номера в порядке возрастания.

Тип алгоритма: Циклический

Текст программы:

```

#Дан целочисленный список размера N, содержащий ровно два одинаковых элемента.
#Найти номера одинаковых элементов и вывести эти номера в порядке возрастания.

def possitive(arr): 1 usage new *
    # Создаем словарь для хранения индексов каждого элемента
    indecs = {}

    for i, num in enumerate(arr):
        if num in indecs:
            # Если элемент уже встречался, возвращаем его индексы
            return sorted([indecs[num], i])
        else:
            # Если элемент встречается впервые, запоминаем его индекс
            indecs[num] = i

    return None # Если одинаковых элементов не найдено

# Пример использования
try:
    N = int(input("Введите размер списка: "))
    if N <= 0:
        print("Ошибка: Размер списка должен быть положительным целым числом.")
    else:
        arr = []
        for i in range(N):
            value = int(input(f"Введите элемент {i + 1}: "))
            arr.append(value)

        result = possitive(arr)
        if result:
            print("Индексы одинаковых элементов:", result)
        else:
            print("Одинаковых элементов не найдено.")
except ValueError:
    print("Ошибка: Пожалуйста, введите целое число.")

```

Протокол программ:

Введите размер списка: 5

Введите элемент 1: 1

Введите элемент 2: 2

Введите элемент 3: 3

Введите элемент 4: 1

Введите элемент 5: 5

Индексы одинаковых элементов: [0, 3]

Постановка задачи:

3. Дан список A размера N и целое число K ($1 < K < 4$, $K < N$). Осуществить циклический сдвиг элементов списка вправо на K позиций (при этом A_1 перейдет в A_{K+1} , A_2 — в A_{K+2} , ..., A_N — в A_K). Допускается использовать вспомогательный список из 4 элементов.

Тип алгоритма: Циклический

Текст программы:

```

1  #Дан список A размера N и целое число K (1 < K < 4, K < N).
2  # Осуществить циклический сдвиг элементов списка вправо на K позиций
3  # (при этом A1 перейдет в AK+1, A2 — в AK+2, ..., AN — в AK).
4  # Допускается использовать вспомогательный список из 4 элементов.
5
6  def possitive(A, K):
7      N = len(A)
8
9      # Проверка на корректность
10     if K < 1 or K >= 4 or K >= N:
11         return None # Возвращаем None при некорректном K
12
13     # Вспомогательный список для хранения последних K элементов
14     aux = [0] * K
15
16
17     for i in range(K):
18         aux[i] = A[N - K + i]
19
20     # Сдвиг элемента вправо
21     for j in range(N - 1, K - 1, -1):
22         A[j] = A[j - K]
23
24     # Копируем последние K элементов в начало списка
25     for i in range(K):
26         A[i] = aux[i]
27
28     return A
29
30
31 N = input("Введите размер списка N: ")
32 A = []
33
34 try:
35     N = int(N)
36     if N <= 0:
37         print("Ошибка: Размер списка должен быть положительным целым числом.")
38     else:
39         # Заполнение списка A
40         for i in range(N):
41             value = int(input(f"Введите элемент A[{i + 1}]: "))
42             A.append(value)
43
44     K = input("Введите значение K (1 < K < 4, K < N): ")
45     K = int(K)
46
47     # Выполнение циклического сдвига
48     result = possitive(A, K)
49
50     if result is None:
51         print("Ошибка: Неверное значение K. Убедитесь, что 1 < K < 4 и K < N.")
52     else:
53         # Вывод результата
54         print("Список после циклического сдвига:", result)
55
56 except ValueError:
57     print("Ошибка: Пожалуйста, введите допустимые целые числа.")

```

Протокол программ:

Введите размер списка N: 4

Введите элемент A[1]: 1

Введите элемент A[2]: 2

Введите элемент A[3]: 3

Введите элемент A[4]: 4

Введите значение K (1 < K < 4, K < N): 2

Список после циклического сдвига: [3, 4, 1, 2]

Process finished with exit code 0

Вывод:

В процессе работы я закрепил полученные ранее навыки, приборел

новые навыки в использование циклов и функций научился создавать программы с использованием циклов и функций, приобрел навыки составление программ со списками в IDE PyCharm Community.