

## Практическое задание № 6.

**Тема:** составление программ со списками в IDE PyCharm Community.

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

### Постановка задачи

Вариант 4

1. Дан первый член  $A$  и знаменатель  $D$  геометрической прогрессии. Сформировать и вывести список размера 10, содержащий 10 первых членов данной прогрессии:  $A, A \cdot D, A \cdot D^2, A \cdot D^3, \dots$ .
2. Дан список размера  $N$ . Найти номер его последнего локального максимума (локальный максимум — это элемент, который больше любого из своих соседей).
3. Дан список размера  $N$ . Переставить в обратном порядке элементы списка, расположенные между его минимальным и максимальным элементами, включая минимальный и максимальный элементы.

**Тип алгоритма:** циклический.

**Текст программы:**

```
1  # Дан первый член A и знаменатель D геометрической прогрессии. Сформировать и
2  # вывести список размера 10, содержащий 10 первых членов данной прогрессии: A,
3  # A* D, A* D 2 , A*D 3, ... .
4
5  a = input("Введите стартовое число для последовательности: ")
6  while type(a) != int:
7      try:
8          a = int(a)
9      except:
10         a = input("Введите стартовое число для последовательности: ")
11  d = input("Введите знаменатель последовательности: ")
12  while type(d) != int:
13      try:
14          d = int(d)
15      except:
16         d = input("Введите знаменатель последовательности: ")
17  print([a*d**(i) for i in range(10)])
```

```

1  # Дан список размера N. Найти номер его последнего локального максимума
2  # (локальный максимум — это элемент, который больше любого из своих соседей).
3
4  import random
5  def findLMax(lst):
6      n = len(lst)
7      if n == 1: return 0 # Единственный элемент автоматически является локальным максимумом
8
9      lastIndex = None # Переменная для хранения индекса последнего локального максимума
10     if lst[0] > lst[1]: lastIndex = 0
11
12     # Проверка элементов с 1 до n-2
13     for i in range(1, n - 1):
14         if lst[i] > lst[i - 1] and lst[i] > lst[i + 1]:
15             lastIndex = i
16
17     # Проверка последнего элемента
18     if lst[-1] > lst[-2]: lastIndex = n - 1
19     return lastIndex
20
21 n = input("Введите длину случайного списка: ")
22 while type(n) != int:
23     try:
24         n = int(n)
25     except:
26         n = input("Введите длину случайного списка: ")
27 lst = [int(random.randint(1, 100)) for i in range(n)]
28 lMax = findLMax(lst)
29 print("Список {}\nНомер последнего локального максимума: {}. Это число {}".format(lst, lMax, lst[lMax]))

```

```

1  # Дан список размера N. Переставить в обратном порядке элементы списка,
2  # расположенные между его минимальным и максимальным элементами, включая
3  # минимальный и максимальный элементы.
4
5  import random
6  def reverseBetweenMinMax(lst):
7      if len(lst) < 2: # Если список слишком короткий (1 или 0 элементов), возвращаем его как есть
8          return lst
9
10     # Находим индексы минимального и максимального элементов и упорядочиваем их
11     start, end = sorted((lst.index(min(lst)), lst.index(max(lst))))
12
13     # Разворачиваем подсписок между минимальным и максимальным элементами (включительно)
14     lst[start:end + 1] = lst[start:end + 1][::-1]
15     return lst
16
17 n = input("Введите длину случайного списка: ")
18 while type(n) != int:
19     try:
20         n = int(n)
21     except:
22         n = input("Введите длину случайного списка: ")
23 lst = [int(random.randint(1, 100)) for i in range(n)]
24 print("Исходный список:", lst)
25 print("Результат:", reverseBetweenMinMax(lst))

```

**Протокол работы программы:**

Введите стартовое число для  
последовательности: 3

Введите знаменатель последовательности: 2  
[3, 6, 12, 24, 48, 96, 192, 384, 768, 1536]

Введите длину случайного списка: 5  
Список [46, 8, 57, 90, 88]

Номер последнего локального максимума: 3.  
Это число 90

Введите длину случайного списка: 5  
Исходный список: [69, 29, 35, 40, 20]  
Результат: [20, 40, 35, 29, 69]

Process finished with exit code 0.

**Вывод:** в процессе выполнения практического задания закрепили усвоенные знания, понятия и алгоритмы, а также основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community и составили программы со списками. Были использованы языковые конструкции `def`, `return`, `input`, `try`, `except`, `sorted`, `while`, `int`, `print`.

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

Готовые программные коды выложены на GitHub.