

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

Постановка задачи 1.

Разработать программу, которая сформирует и выведет список размера 10, содержащий 10 первых членов данной прогрессии: $A, A \cdot D, A \cdot D^2, A \cdot D^3, \dots$

Тип алгоритма: циклический.

Текст программы:

```
# Дан первый член A и знаменатель D геометрической прогрессии.
# Сформировать и вывести список размера 10, содержащий
# 10 первых членов данной прогрессии: A, A * D, A * D2, A * D3, ...

# создаем функцию, в которой заполняем список прогрессией
def listik(a, d):
    cd = []
    i = 0
    while i < 10:
        cd.append(a * d ** i)
        i += 1
    return cd

# просим первый член прогрессии и множитель
print('Первые 10 членов прогрессии: ',
      listik(int(input('Введите первый член прогрессии: ')),
             int(input('Введите множитель: '))))
```

Протокол работы программы:

Введите первый член прогрессии: 1

Введите множитель: 3

Первые 10 членов прогрессии: [1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683]

Process finished with exit code 0

Постановка задачи 2.

Разработать программу, которая находит номер последнего локального максимума в списке размера N

Тип алгоритма: циклический.

Текст программы:

```
# Дан список размера N. Найти номер его последнего
# локального максимума (локальный максимум — это элемент,
# который больше любого из своих соседей).

import random

# создаем список и находим последний локальный максимум
def local_max(n):
    spisok = [random.randint(0,10) for b in range(n)]
```

```
max = []
for i in range(0, n-1):
    if spisok[i] > spisok[i-1]:
        if spisok[i] > spisok[i+1]:
            max = spisok[i]
return spisok, max

# просим размер списка и выводим список и последний локал макс
print('Созданный список и его последний локальный максимум: ',
      local_max(int(input('Введите размер списка: '))))
```

Протокол работы программы:

Введите размер списка: 15

Созданный список и его последний локальный максимум: ([8, 9, 2, 9, 5, 6, 10, 2, 9, 2, 8, 8, 7, 0, 8], 9)

Process finished with exit code 0

Постановка задачи 3.

Разработать программу, которая переставит в обратном порядке элементы списка N, расположенные между его минимальным и максимальным элементами, включая минимальный и максимальный элементы.

Текст программы:

```
# Дан список размера N. Переставить в обратном порядке
# элементы списка, расположенные между его минимальным
# и максимальным элементами, включая минимальный
# и максимальный элементы.

import random

# создаем функцию, в котором работаем со списком и его мин и макс
def reverse_spisok(n):
    spisok = (random.sample(range(15), n))

    n1 = min(spisok)
    n2 = max(spisok)

    n1_id = spisok.index(n1)
    n2_id = spisok.index(n2)

    n1_id, n2_id = min(n1_id, n2_id), max(n1_id, n2_id)

    rev_spisok = spisok[0:n1_id]+spisok[n1_id:n2_id+1][::-1]+spisok[n2_id+1:]

    return f'Изначальный список: {spisok}\nМинимум и максимум: {n1} {n2}\nПеревёрнутый список: {rev_spisok}'

# запрашиваем число и выводим списки и мин макс
print(reverse_spisok(int(input('Введите размер списка: '))))
```

Протокол работы программы:

Введите размер списка: 8

Изначальный список: [14, 11, 10, 8, 7, 2, 9, 6]

Минимум и максимум: 2 14

Перевернутый список: [2, 7, 8, 10, 11, 14, 9, 6]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community.

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

Готовые программные коды выложены на GitHub