

Практическое задание № 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 первый член A и разность D арифметической прогрессии. Сформировать и вывести список размера 10, содержащий 10 первых членов данной прогрессии: A , $A + D$, $A + 2*D$, $A + 3*D$,

Тип алгоритма: циклический.

Текст программы:

```
# Дан первый член A и разность D арифметической прогрессии.
# Сформировать и
# вывести список размера 10, содержащий 10 первых членов данной
# прогрессии: A, A
# + D, A + 2*D, A + 3*D, ... .

a = input("Введите стартовое число арифметической прогрессии: ")
while type(a) != int: # обработка исключений
    try:
        a = int(a)
    except ValueError:
        print("Неправельно ввели!")
        a = input("Введите стартовое число арифметической прогрессии: ")

d = input("Введите разность арифметической прогрессии: ")
```

```
while type(d) != int: # обработка исключений
    try:
        d = int(d)
    except ValueError:
        print("Неправельно ввели!")
        d = input("Введите разность арифметической прогрессии: ")

ar_pr = []
for i in range(10):
    ar_pr.append(a + d * i)

print(ar_pr)
```

Протокол работы программы:

Введите стартовое число арифметической прогрессии: 10

Введите разность арифметической прогрессии: 2

[10, 12, 14, 16, 18, 20, 22, 24, 26, 28]

Process finished with exit code 0.

Постановка задачи №2.

Дан список размера N. Найти номер его первого локального минимума (локальный минимум — это элемент, который меньше любого из своих соседей).

Тип алгоритма: циклический.

Текст программы:

Дан список размера N. Найти номер его первого локального минимума (локальный

минимум — это элемент, который меньше любого из своих соседей).

```
import random
```

```
def find_l_min(lst):
```

```
    if len(lst) == 1:
```

```
        loc_min = 0
```

```
        return loc_min
```

```
    # Проверка первого числа
```

```
    if lst[0] < lst[1]:
```

```
        loc_min = 0
```

```
        return loc_min
```

```
    # Проверка эл-тов с 1 до предпоследнего
```

```
    for i in range(1, n - 1):
```

```
        if lst[i] < lst[i - 1] and lst[i] < lst[i + 1]:
```

```
            loc_min = i
```

```
            return loc_min
```

```
    # Проверка последнего эл-та
```

```
    if lst[-1] < lst[-2]:
```

```
        loc_min = n - 1
```

```
        return loc_min
```

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

spisoc = []
i = 0
while i < n:
    spisoc.append(random.randint(0, 100))
    i += 1

l_min = find_l_min(spisoc)
print("Список:", spisoc)
print("Номер первого локального минимума списка:", l_min)
```

Протокол работы программы:

Введите длину списка: 13

Список: [62, 36, 96, 65, 38, 72, 69, 80, 5, 37, 44, 13, 44]

Номер первого локального минимума списка: 1

Process finished with exit code 0.

Постановка задачи №3.

Дан список размера N (N — четное число). Поменять местами его первый элемент со вторым, третий — с четвертым и т. д.

Тип алгоритма: циклический.

Текст программы:

```
import random

n = input("Введите длину списка (число должно быть чётным): ")
while type(n) != int: # обработка исключений
    try:
        n = int(n)
        while n % 2 != 0: # приведение к чётному числу
            print("Длина списка должна быть чётной!")
            n = input("Введите длину списка (число должно быть чётным): ")
            while type(n) != int: # обработка исключений
                try:
                    n = int(n)
                except ValueError:
                    print("Неправильно ввели!")
                    n = input("Введите длину списка (число должно быть чётным): ")
        except ValueError:
            print("Неправильно ввели!")
            n = input("Введите длину списка (число должно быть чётным): ")

spisoc = []
i = 0
while i < n:
    spisoc.append(random.randint(0, 100))
    i += 1
```

```
print("Исходный список:", spisoc)
for i in range(0, len(spisoc), 2):
    spisoc[i], spisoc[i + 1] = spisoc[i + 1], spisoc[i]
print("Список после обмена:", spisoc)
```

Протокол работы программы:

Введите длину списка (число должно быть чётным): 677

Длинна списка должна быть чётной!

Введите длину списка (число должно быть чётным): 24

Исходный список: [73, 56, 83, 96, 0, 22, 54, 66, 43, 91, 4, 8, 34, 19, 97, 68, 67, 99, 87, 95, 23, 24, 19, 82]

Список после обмена: [56, 73, 96, 83, 22, 0, 66, 54, 91, 43, 8, 4, 19, 34, 68, 97, 99, 67, 95, 87, 24, 23, 82, 19]

Process finished with exit code 0.

Вывод: в процессе выполнения практического задания я закрепила усвоенные знания, понятия и алгоритмы, а также основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community и составила программы со списками. Были использованы языковые конструкции input, try, except, def, return, while, int, if, print, for, import.

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

Готовые программные коды выложены на GitHub.