

## 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 целочисленный список A размера 10. Вывести порядковый номер последнего из тех его элементов  $A_k$ , которые удовлетворяют двойному неравенству  $A_1 < A_k < A_{10}$ . Если таких элементов нет, то вывести 0.
2. Дано число R и список размера N. Найти два различных 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).
3. Дан список размера N, все элементы которого, кроме первого, упорядочены по возрастанию. Сделать список упорядоченным, переместив первый элемент на новую позицию.

**Текст программы:**

**1.**

# Дан целочисленный список A размера 10. Вывести порядковый номер последнего из тех его элементов  $A_k$ , которые  
# удовлетворяют двойному неравенству  $A_1 < A_k < A_{10}$ . Если таких элементов нет, то вывести 0.

```
import random

spisok = []
i = 0
while i < 10: # В цикле while создаём список размера 10 с randomным набором
    элементов
    spisok.append(random.randint(0, 100))
    i += 1
print(spisok)

num1 = spisok[0] # Присваивание num1 значение первого элемента списка
num2 = spisok[9] # Присваивание num2 значение последнего элемента списка
s = 0
for elem in spisok:
    if num1 < elem < num2: # Если элемент списка соответствует условию, то
        его значение присваивается s, иначе 0
        s = elem
if s == 0:
    print('В списке нет чисел, удовлетворяющих условию: 0')
else:
    k = spisok.index(s) # Получение индекса числа, удовлетворяющего условию
    print('Индекс последнего числа, удовлетворяющего условию: ', k)
```

**2.**

```
# Дано число R и список размера N. Найти два различных 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

r = int(input('Введите число R: '))
n = int(input('Введите длину списка: '))
spisok = []
i = 0

while i < n:
    spisok.append(random.randint(0, 100)) # В цикле while создаём список
    i += 1
    # размера n с randomным набором элементов
spisok.sort() # Сортируем список, чтобы индексация в spisok и последующем
# списке lit совпадала
print('Исходный список: ', spisok)

lit = []
for elem in spisok: # Создаём новый список lit, куда добавляем разность |Ak
# - R | для определения близости элементов
    razn = elem - r # в списке к числу R. Отрицательные делаем
    # положительными, т.к. при сравнении они будут
    if razn < 0: # давать неправильный результат.
        razn *= (-1)
    lit.append(razn)
print('Список разности Ak - R: ', lit)
a = min(lit) # Ищем два наименьших элемента из списка с разностью числа
# Ak и R, и находим их индексы, чтобы потом
c = lit.index(a) # впоследствии обратиться по индексам к элементам в
# исходном списке
lit.remove(a)
b = min(lit)
lit.append(a) # Возвращаем a, чтобы индексация была правильной, без неё она
# сдвинется на 1 значение
lit.sort()
v = lit.index(b)
num1 = spisok[c]
num2 = spisok[v]

print('Два элемента, наиболее приближённых к числу R: ', num1, num2)
```

### 3

```
# Дан список размера N, все элементы которого, кроме первого, упорядочены по
# возрастанию. Сделать упорядоченным,
# переместив первый элемент на новую позицию.
import random

n = int(input('Введите длину списка: '))
spisok = []
i = 0
while i < n:
    spisok.append(random.randint(0, 100)) # В цикле while создаём список
    i += 1
    # размера n с randomным набором элементов
print('Первоначальный список: ', spisok)
b = spisok[1:]
b.sort()
```

```
spisok[1::] = []  
for elem in b: # В список, содержащий только самый первый элемент, добавляем  
остальные отсортированные элементы списка  
    spisok.append(elem)  
print('Сортировка списка, кроме 1 - го элемента: ', spisok)  
spisok.sort()  
print('Сортировка списка полностью: ', spisok)
```

**Протокол работы программы:**

**1.**

[55, 46, 50, 69, 39, 97, 53, 98, 72, 96]

Индекс последнего числа, удовлетворяющего условию: 8

Process finished with exit code 0

*Второй случай*

[21, 42, 84, 9, 27, 100, 27, 29, 58, 10]

В списке нет чисел, удовлетворяющих условию: 0

Process finished with exit code 0

**2.**

Введите число R: 50

Введите длину списка: 4

Исходный список: [36, 41, 55, 85]

Список разности  $A_k - R$ : [14, 9, 5, 35]

Два элемента, наиболее приближённых к числу R: 55 41

Process finished with exit code 0

**3.**

Введите длину списка: 5

Первоначальный список: [25, 13, 49, 23, 17]

Сортировка списка, кроме 1 - го элемента: [25, 13, 17, 23, 49]

Сортировка списка полностью: [13, 17, 23, 25, 49]

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия выработала навыки составления программ с использованием списков в IDE PyCharm Community. Были использованы языковые конструкции `for`, `while`.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.