

Лабораторная работа №4: «Одномерные массивы»

Цель работы:

Дать студентам практический навык в написании программ обработки одномерных массивов: поиск максимумов и минимумов, сортировка.

Постановка задачи

Сформировать одномерный список, состоящий из N вещественных чисел, полученных генератором случайных чисел. Количество элементов списка (N) запрашивается у пользователя, но не превышает 30. Диапазон значений элементов от -5.0 до 5.0. Вычислить:

1. Количество элементов массива, больших C
2. Произведение элементов, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные нулю, считать положительными).

Теоретическое введение

Для работы с одномерными массивами и матрицами (многомерными массивами) в Python имеются специализированные модули и библиотеки (`array`, `numpy`, ...).

Массив - это конечная именованная последовательность однотипных величин.

Для организации такой структуры в Python можно использовать такие структуры данных, как списки, кортежи, множества и диапазоны, которые представляют нумерованные наборы объектов. Каждый элемент набора содержит ссылку на объект, который, в свою очередь, может представлять объект произвольного типа данных и иметь неограниченную степень вложенности.

В решении этого задания для хранения однотипных данных (массивов данных) предлагается использовать структуру данных, которая называется список.

Список представляет собой последовательность элементов, пронумерованных от 0 (нуля). Элементы списка могут иметь различные типы. Список можно задать перечислением элементов списка в квадратных скобках, например, список можно задать так:

```
Preshe = [1, 3, 5, 7, 11, 13]
```

```
PColor = ['Red', 'Orange', 'Yellow', 'Green', 'Blue']
```

или так:

```
Trest = [1, 3, 'Old', 7, 'Or', 13]
```

В списке `Preshe` – 6 элементов, а именно `Preshe[0] == 1`, `Preshe[1] == 3`, `Preshe[2] == 5`, `Preshe[3] == 7`, `Preshe[4] == 11`, `Preshe[5] == 13`.

Список `PColor` состоит из 5 элементов, каждый из которых является строкой, а вот список `Trest` состоит из шести элементов, часть которых - число, а часть - строка.

При обращении к элементам списка можно использовать как положительные индексы, так и отрицательные. Если индекс положительный, то счет ведется от нуля до максимального элемента, слева направо. Если индекс отрицательный, то счет ведется справа налево: `Preshe[-1] == 13`, `Preshe[-6] == 1`.

Количество элементов в списке (длину списка), можно получить при помощи функции `len`, например, `len(Preshe) == 6`.

Существует несколько способов работы со списком. Разработчики предлагают различные варианты от специальных модулей до библиотек. Стандартные решения языка для нашего примера достаточны.

Рассмотрим один из способов создания списка и его наполнения:

- Запросить размер списка у пользователя. Пусть этот размер не должен быть меньше 5 элементов и не превышать 30;
- Создать пустой список (не содержащий элементов, длины 0);
- Наполним список случайными числами, применяя метод `append`.

```
n = int(input("Элементов в списке (N<=30) N: "))
if n > 30: n = 30
elif n < 5: n = 5
mas = [] # Создаем пустой список
for i in range(n): # Инициализация
    mas.append(uniform(-5, 5))
```

Если использовать, например, модуль `array`, то изменения в программе будут минимальными:

```
# создание массива нулевой длины
# с элементами вещественного типа
mas = array('f')
for i in range(n): # Инициализация
    mas.append(uniform(-5, 5))
```

В решении задачи используется цикл с параметром, что позволяет получать доступ к элементам массива:

```
for <Текущий элемент> in <Последовательность>:
    <Инструкции внутри цикла>
```

Для обмена с консолью (ввод/вывод) использованы стандартные функции `input()` и `print()`.

После формирования списка, в следующем цикле подсчитывается сумма элементов, модуль которых не превышает единицу. В этом же цикле при обнаружении в

массиве элемента, значение которого превышает пороговое значение, выполняется обнуление элемента.

Поиск первого и второго максимальных элементов построен по принципу однопроходного алгоритма.

Модуль элемента $A[i]$ сравнивается с первым максимальным элементом $Max1$.

Если $abs(A[i]) > Max1$, то значение $Max1$ сдвигается на позицию второго максимального элемента ($Max2$), а $Max1 = abs(A[i])$. Иначе модуль элемента $A[i]$ сравнивается со вторым максимальным элементом $Max2$.

Для сжатия массива по заданному принципу (нулевые элементы размещаются в конце массива при сохранении порядка ненулевых элементов), используется следующий алгоритм: используется дополнительный индекс j . Массив просматривается с первого элемента. При обнаружении не нулевого элемента он копируется в элемент с индексом j и индекс j инкрементируется. После просмотра массива все элементы, начиная с элемента с индексом j обнуляются.

Описание алгоритма

1. Запросить количество элементов N и пороговое значение $Amax$.
2. Инициализировать массив случайными данными и вывести начальное состояние.
3. В цикле от 0 до $N-1$. Найти сумму элементов, модуль которых меньше 1, и обнулить элементы, значение которых превысило установленный порог $Amax$.
4. Инициализировать $Max1$ и $Max2$ модулем значения нулевого элемента массива.
5. В цикле от 1 до $N-1$. Если модуль элемента массива больше $Max1$, то $Max1$ сохранить в $Max2$, а модуль элемента массива в $Max1$. Иначе, если модуль элемента массива больше $Max2$, то модуль элемента массива сохранить в $Max2$.
6. Инициализировать переменную j .
7. В цикле от 0 до $N-1$. Если значение элемента больше нуля, то копировать в элемент с индексом j . Увеличить j на 1.
8. В цикле от j до $N-1$. Все элементы приравнять нулю.
9. Вывести полученный массив и значения $Max1$, $Max2$ и суммы.

Описание входных и выходных данных

В предшествующей работе был принят вещественный тип данных (*real*). В этой работе он сохранён.

Листинг программы

```
# -*- coding: UTF-8 -*-

from random import uniform
count = int(input("input count of elements N (N <= 30): "))
c = float(input("input C: "))
mas = []
for i in range(0, count):
    mas.append(uniform(-5.0, 5.0))
print("начальное состояние\n{0}".format(mas))
count2 = 0
```

```

for i in range(0, count):
    if mas[i] > c:
        count2 += 1
print("Count of elements bigger then C: {0}".format(count2))
bn = -1
for i in range(0, count):
    if abs(mas[i]) > bn:
        bn = abs(mas[i])
        pos = i
print("Numbers after biggest module: ", end=' ')
for i in range(pos, count):
    print("{0} ".format(mas[i]), end=' ')
print()
for i in range(0, count-1):
    for j in range(i, count):
        if mas[i] > mas[j]:
            prom = mas[i]
            mas[i] = mas[j]
            mas[j] = prom
print("Конечное состояние\n{0}".format(mas))

```

Результат работы программы

input count of elements N (N <= 30): 10

input C: -0.1

начальное состояние

```

[-0.3797388956712009, 3.8596895402545144, 2.6043898718916667, \
-1.9475674797300355, -0.21472689580796533, -2.443928363347434, \
-3.1226520126748527, 2.9352987963320523, 0.3204063757130706, \
-0.29930675406810536]

```

Count of elements bigger then C: 4

```

Numbers after biggest module: 3.8596895402545144 \
2.6043898718916667 -1.9475674797300355 -0.21472689580796533 \
-2.443928363347434 -3.1226520126748527 2.9352987963320523 \
0.3204063757130706 -0.29930675406810536

```

Конечное состояние

```

[-3.1226520126748527, -2.443928363347434, -1.9475674797300355, \
-0.3797388956712009, -0.29930675406810536, -0.21472689580796533, \
\ 0.3204063757130706, 2.6043898718916667, 2.9352987963320523, \
3.8596895402545144]

```

Задание к лабораторной работе №4 «Одномерные массивы»

Вариант 1

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Сумму отрицательных элементов.
 2. Произведение элементов, расположенных между максимальным и минимальным элементами.
- Упорядочить элементы массива по возрастанию.

Вариант 2

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Сумму положительных элементов.
 2. Произведение элементов, расположенных между максимальным по модулю и минимальным по модулю элементами.
- Упорядочить элементы массива по убыванию.

Вариант 3

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

1. Произведение элементов с четными номерами.
2. Сумму элементов, расположенных между первым и последним нулевыми элементами. Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом - все отрицательные (элементы, равные нулю, считать положительными).

Вариант 4

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Сумму элементов с нечетными номерами.
2. Сумму элементов, расположенных между первым и последним отрицательными элементами. Сжать массив, удалив из него все элементы, модуль которых не превышает единицу. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 5

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Максимальный элемент массива.
 2. Сумму элементов, расположенных до последнего положительного элемента.
- Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$.

Освободившиеся в конце массива элементы заполнить нулями.

Вариант 6

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Минимальный элемент массива.

2. Сумму элементов, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом - все остальные.

Вариант 7

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

1. Номер максимального элемента массива.

2. Произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине - элементы, стоявшие в четных позициях.

Вариант 8

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Номер минимального элемента.

2. Сумму элементов, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает единицу, а потом - все остальные.

Вариант 9

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Максимальный по модулю элемент.

2. Сумму элементов, расположенных между первым и вторым положительными элементами. Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Вариант 10

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

1. Минимальный по модулю элемент.

2. Сумму модулей элементов, расположенных после первого элемента, равного нулю. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине - элементы, стоявшие в нечетных позициях.

Вариант 11

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Номер минимального по модулю элемента.

2. Сумму модулей элементов, расположенных после первого отрицательного элемента. Сжать массив, удалив из него все элементы, величина которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 12

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Номер максимального по модулю элемента.

2. Сумму элементов, расположенных после первого положительного элемента. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a, b]$, а потом все остальные.

Вариант 13

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Количество элементов массива, лежащих в диапазоне от A до B .
 2. Сумму элементов, расположенных после максимального элемента.
- Упорядочить элементы массива по убыванию модулей.

Вариант 14

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Количество элементов массива, равных нулю.
2. Сумму элементов, расположенных после минимального элемента. Упорядочить элементы массива по возрастанию модулей.

Вариант 15

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. Количество элементов массива, больших C .
2. Произведение элементов, расположенных после максимального по модулю элемента. Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные нулю, считать положительными).