

Лабораторная работа №4: "Одномерные массивы"

Цель работы

Дать студентам практический навык в написании программ обработки одномерных массивов: поиск максимумов и минимумов, сортировка².

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. Первый и второй максимальные по модулю элементы списка.
2. Сумму элементов, модуль которых меньше единицы.
3. Все элементы, модуль которых превышает A_{max} обнулить.
3. Отсортировать список, сохраняя порядок ненулевых элементов. Равные нулю элементы разместить в конце списка.

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`.

² Структуры данных Python, могут использоваться для организации массивов. В этих структурах могут быть методы и функции поиска максимума, минимума или сортировки. В этом руководстве предлагается использовать алгоритмы поиска решений без использования готовых методов и функций.

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 A_{max} .
2. Инициировать 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*.
5. Инициализировать переменную *j*.
6. В цикле от 0 до N-1. Если значение элемента больше нуля, то копировать в элемент с индексом *j*. Увеличить *j* на 1.
7. В цикле от *j* до N-1. Все элементы приравнять нулю.
8. Вывести полученный массив и значения *Max1*, *Max2* и суммы.

Описание входных и выходных данных

Поскольку тип элементов массива задан как вещественный, то тип переменных, используемых в подсчётах, также определим как вещественный (**float**).

Листинг программы

```
# -*- coding: cp1251 -*-
from math import *
from random import *
n = int(input("Элементов в массиве (N<=30) N: "))
if n > 30: n = 30
elif n < 5: n = 5
amax = float(input("Пороговое значение A: "))
# Генерация массива и вывод
print("Начальное состояние")
mas = []
for i in range(n):
    mas.append(uniform(-5, 5))
    print("{0: 7.3f}".format(mas[i]), end=" ")
print()
# Нахождение суммы
# Обнуление элементов превысивших порог
asum = 0.0
for i in range(n): # от 0 до n-1
    if abs(mas[i]) < 1.0:
        asum = asum + mas[i]
    if abs(mas[i]) > amax:
        mas[i] = 0.0
# Поиск максимального элемента
max1 = abs(mas[0])
max2 = abs(mas[0])
for i in range(1,n):
    if max1 < abs(mas[i]):
        max2 = max1
        max1 = abs(mas[i])
    else:
        if max2 < abs(mas[i]):
            max2 = abs(mas[i])
# Сортировка массива
j = 0
for i in range(n):
```

```

        if abs(mas[i]) > 0.00001:
            mas[j] = mas[i]
            j = j + 1
    for i in range(j,n): # от j до n-1
        mas[i] = 0.0
    # Массив после сортировки
    print("Конечное состояние")
    for i in range(n):
        print("{0: 7.3f}".format(mas[i]), end=" ")
    print()
    print("max1={0: 7.3f} max2={1: 7.3f} sum={2: 7.3f}" \
          .format(max1, max2, asum))

```

Результат работы программы

```

Элементов в массиве(N<=30) N: 7
Пороговое значение A: 2.8
Начальное состояние
-2.655   2.194   1.034   2.804  -1.490  -0.474  -0.817
Конечное состояние
-2.655   2.194   1.034  -1.490  -0.474  -0.817   0.000
max1=  2.655 max2=  2.655 sum= -1.291

```

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).

Вариант 16

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

1. Количество отрицательных элементов.
2. Сумму модулей элементов, расположенных после минимального по модулю элемента. Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

Вариант 17

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

1. Количество положительных элементов.
2. Сумму элементов, расположенных после последнего элемента, равного нулю. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает единицу, а потом - все остальные.

Вариант 18

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

1. Количество элементов массива, меньших C .
2. Сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20 %, а потом - все остальные:

Вариант 19

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

1. Произведение отрицательных элементов.
 2. Сумму положительных элементов, расположенных до максимального элемента.
- Изменить порядок следования элементов в массиве на обратный.

Вариант 20

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

1. Произведение положительных элементов.
 2. Сумму элементов, расположенных до минимального элемента.
- Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

Вариант 21

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

1. Максимальный элемент массива.
2. Сумму остатков получаемых от деления элементов массива на максимальный элемент. Упорядочить элементы массива по возрастанию остатков, полученных от их деления на максимальный элемент.

Вариант 22

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

1. Среднее значение, как отношение суммы всех элементов на их количество $\bar{a} = \frac{1}{n} \cdot \sum_{i=1}^n a_i$.
2. Сумму квадратов разностей между элементами массива и полученным средним значением $S = \sum (a_i - \bar{a})^2$.

Упорядочить элементы массива по убыванию их разности со средним значением $(a_i - \bar{a})$: вначале расположить элементы с положительной максимальной разностью, а затем остальные по убыванию разности.

Вариант 23

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

1. Второй по величине элемент.
2. Сумму элементов, расположенных между максимальным и вторым по величине элементами.

Упорядочить элементы массива по возрастанию остатков, полученных от их деления на второй по величине элемент.

Вариант 24

В одномерном массиве, состоящем из n вещественных элементов:

1. Вычислить максимальный и минимальный элементы (Max и Min).
2. Выполнить вставку нового значения в элемент, который расположен в середине между максимальным и минимальным элементами. Вставку выполнить следующим образом: запросить новое значение, которое должно быть в диапазоне ($Min \leq a \leq Max$); переместить вправо на одну позицию элементы массива от точки вставки (последний элемент теряется); освободившемуся элементу присвоить новое значение.

Упорядочить по возрастанию модулей разности элементов массива и нового значения:

$$|a_i - a|$$

Вариант 25

С использованием модуля *Random* сформировать одномерный массив, состоящий из n вещественных элементов в котором элементы принимают случайное значение, и выполняется условие: $a_1 < a_2 < a_3 < \dots < a_n$.

1. Для заданного числа y , такого, что $a_1 < y < a_n$, определить такой k , чтобы $a_k < y < a_{k+1}$.
2. Вычислить сумму элементов массива, расположенных до элемента a_k .

Упорядочить по убыванию элементы, стоящие после элемента a_k .

Вариант 26

С использованием модуля *Random* сформировать одномерный массив, состоящий из n вещественных элементов в котором элементы принимают случайное значение, и выполняется условие: $a_1 > a_2 > a_3 > \dots > a_n$.

1. Для заданного числа y , такого, что $a_1 < y < a_n$, определить такой k , чтобы $a_k < y < a_{k+1}$.
2. Вычислить сумму элементов массива, расположенных после элемента a_k .

Упорядочить по убыванию элементы, стоящие до элемента a_k .

Вариант 27

С использованием модуля *Random* сформировать одномерный массив, состоящий из n вещественных элементов в котором элементы случайным образом принимают положительный или отрицательный знак и значение от -5 до 5. Для заданного числа y , такого, что $a_{min} < y < a_{max}$, вычислить:

1. Сумму элементов массива, значения модуля которых меньше y .
2. Произведение остальных элементов.

Вариант 28

С использованием модуля *Random* сформировать одномерный массив, состоящий из n вещественных элементов в котором элементы случайным образом принимают положительный или отрицательный знак и значение от -10 до 10. Для заданного числа y , такого, что $a_{min} < y < a_{max}$, вычислить:

1. Произведение элементов массива, значения модуля которых больше y .
2. Сумму модулей остальных элементов.

Вариант 29

Координаты n точек заданы как элементы одномерного массива. Нечётные элементы - значения ординат, а чётные - абсцисс. Вычислить максимальное расстояние между ближайшими точками.

Вывести координаты всех точек, которые попадают в круг с координатами центра X_0, Y_0 и радиусом R .

Упорядочить положение точек по возрастанию их ординат.

Вариант 30

Координаты n точек заданы как элементы одномерного массива. Нечётные элементы - значения ординат, а чётные - абсцисс. Вычислить минимальное расстояние между ближайшими точками.

Вывести координаты всех точек, которые попадают в кольцо с координатами центра X_0, Y_0 , внутренним радиусом R_1 и внешним радиусом R_2 .

Упорядочить положение точек по возрастанию их абсцисс.