

Практические задания

1. Написать программу проверяющую наличие хотя бы одного отрицательного числа в целочисленном списке и выводящую сообщение о том, есть такое число или нет.

Ввод: список целых чисел.

Вывод: результат проверки.

Примеры:

| Ввод | Вывод |
|-----------------------------|--------------------------|
| [4, -7, -1, 8, -2, 9, 1, 6] | Отрицательное число есть |
| [5, 9, 3, 2, 1, 8, 4, 3, 6] | Отрицательного числа нет |

2. Дан список из N случайных целых чисел из диапазона от -10 до 10. Отсортировать в нем по возрастанию все положительные элементы. Неположительные должны остаться на своих местах.

Ввод: число N – количество элементов списка.

Вывод: первоначальный список и список, отсортированный по условию задачи.

Пример:

| Ввод | Вывод |
|------|--|
| 8 | 4 -7 -1 8 -2 9 1 6 1 -7 -1 4 -2 6 8 9 |

3. Дан список из N случайных чисел (диапазон любой). Вывести сам список. Затем вывести все различные элементы списка. Повторяющиеся выводить не нужно.

Ввод: число N – количество элементов списка.

Вывод: 2 списка.

Пример:

| Ввод | Вывод |
|------|--------------------------------------|
| 10 | 7 8 2 9 8 2 1 3 5 8 7 8 2 9 1 3 5 |

4. Составить программу, создающую список чисел Фибоначчи.

Ввод: количество чисел Фибоначчи.

Вывод: список чисел Фибоначчи.

Пример:

| Ввод | Вывод |
|------|------------------|
| 8 | 0 1 1 2 3 5 8 13 |

5. Проверить уникальность элементов списка. В списке чисел из N элементов проверить, все ли элементы являются уникальными, т.е. каждое число встречается только один раз. Вывести соответствующее сообщение.

Ввод: число N – количество элементов списка.

Вывод: результат проверки.

Примеры:

| Ввод | Вывод |
|------|--------------------------------------|
| 10 | 7 8 2 9 8 2 1 3 5 8 Не уникальный |
| 8 | 2 4 5 7 8 1 3 9 уникальный |

6. Два списка A и B заполнены случайными числами. Сформировать список C , состоящий из попарной суммы элементов с одинаковыми индексами двух первых списков. Размер списка C определяется размером наибольшего из списков A и B . Числа из наибольшего списка, не имеющие пару для образования суммы, записываются в список C без изменений.

Ввод: число N – количество элементов списка.

Вывод: результат проверки.

Примеры:

| Ввод | Вывод |
|---|--------------------------|
| [1, 2, 3, 4, 5] [8, 6, 4, 2, 0, 3, 2, 1] | [9, 8, 7, 6, 5, 3, 2, 1] |
| [1, 2, 3, 4, 5, 6, 7] [-7, -5, -4] | [-6, -3, -1, 4, 5, 6, 7] |

7. Составить программу для нахождения второго по величине элемента списка. Например, в массиве второй по величине элемент: 19 на позиции 7.

Ввод: список случайных целых чисел.

Вывод: значение и позиция второго по величине элемента массива.

Пример:

| Ввод | Вывод |
|--|---------------------------|
| [17, 2, 20, 13, 20, 18, 17, 19, 10, 7] | Элемент: 19 Позиция: 7 |

8. Заполнить при помощи циклов двумерный список следующим образом:

```

1  2  3  4  5
10 9  8  7  6
11 12 13 14 15
20 19 18 17 16

```

Ввод: не требуется.

Вывод: двумерный список в виде таблицы.

9. Написать программу для кольцевого сдвига влево или вправо введённого списка на указанное количество позиций. Если количество позиций – отрицательное число, то сдвиг влево, иначе сдвиг вправо.

Ввод: список и целое число – сдвиг.

Вывод: список, сдвинутый влево или вправо на указанное число позиций.

Пример:

| Ввод | Вывод |
|--------------------------------------|--------------------------------|
| [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] -3 | [3, 4, 5, 6, 7, 8, 9, 0, 1, 2] |
| [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] 3 | [7, 8, 9, 0, 1, 2, 3, 4, 5, 6] |

10. Заполнить при помощи list comprehension двумерный список нулями и единицами в шахматном порядке:

```

1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1

```

Ввод: не требуется.

Вывод: двумерный список в виде таблицы.