**Задача №1**

|  |  |
| --- | --- |
| Ввод | input.txt |
| Вывод | стандартный вывод или output.txt |

В компании подарили каждому сотруднику два талона на обед в близлежащем ресторане. Сотрудники могли взять талоны и записать номер своего бейджика в специальный бланк. Так сделали все кроме одного сотрудника. Нужно вычислить, кто же это такой! На вход подается непустой массив целых чисел. Каждое из них, за исключением одного, встречается 2 раза. Размер массива не превосходит 10000. Числа по модулю не больше 10000. Нужно выяснить, кто не такой, как все!

**Формат ввода**

В первой строке записано число n. Во второй строке через пробел записаны n чисел. Длина массива и каждое из чисел не превосходят 10000.

**Формат вывода**

Выведите единственное число, соответствующее id отличившегося сотрудника

**Пример 1**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| 5  4 1 2 1 2 | 4 |

**Пример 2**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| 5  42 67 67 42 42 | 42 |

**Задача №2**

|  |  |
| --- | --- |
| Ввод | input.txt |
| Вывод | стандартный вывод или output.txt |

Дана скобочная последовательность. Нужно определить, правильная ли она.

Будем придерживаться такого определения:

* пустая строка — правильная скобочная последовательность;
* правильная скобочная последовательность, взятая в скобки одного типа, — правильная скобочная последовательность;
* правильная скобочная последовательность с приписанной слева или справа правильной скобочной последовательностью — тоже правильная.

На вход подается последовательность из скобок трёх видов: [], (), {}.

Напишите функцию isCorrectBracketSeq, которая принимает на вход скобочную последовательность и возвращает True, если последовательность правильная, иначе False.

**Формат ввода**

На вход подается одна строка, содержащая скобочную последовательность.

**Формат вывода**

True или False.

**Пример 1**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| {[()]} | True |

**Пример 2**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| () | True |

**Пример 3**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| (){[()]}() | True |

**Пример 4**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| (){[()][}] | False |

**Задача №3**

|  |  |
| --- | --- |
| Ввод | input.txt |
| Вывод | стандартный вывод или output.txt |

Дана матрица. Нужно написать функцию, которая для элемента возвращает всех его соседей. Соседним считается элемент, находящийся от текущего на одну ячейку влево, вправо, вверх или вниз. Диагональные элементы соседними не считаются.

Например, в матрице A:

[[1,2,3],

[0,2,6],

[7,4,1],

[2,7,0]]

соседними элементами для (0, 0) будут 2 и 0 А для (2, 1) — 1, 2, 7, 7.

**Формат ввода**

В первой строке задано n - количество строк матрицы. Во второй - m - количество столбцов. Числа m и n не превосходят 1000. В следующих n строках задана матрица. Элементы матрицы - целые числа, по модулю не превосходящие 1000. В последних двух строках записаны координаты элемента (индексация начинается с нуля), соседей которого нужно найти.

**Формат вывода**

Напечатайте нужные числа в возрастающем порядке через пробел.

**Пример 1**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| 4  3  1 2 3  0 2 6  7 4 1  2 7 0  3  0 | 7 7 |

**Пример 2**

|  |  |
| --- | --- |
| **Ввод (из input.txt)** | **Вывод** |
| 4  3  1 2 3  0 2 6  7 4 1  2 7 0  0  0 | 0 2 |