**1 задание**

Ограничение времени 1 секунда

Ограничение памяти 64 МБ

Олег — настоящий герой, чьи школьные будни наполнены заботами, уроками и оценками. Он изо всех сил старается, но, как и любой человек, он время от времени допускает ошибки и получает не лучшие оценки.

Сегодня Олег стоит перед особенным испытанием — ему предстоит показать своим родителям свои оценки. Родители попросили показать ему все его оценки за какие-то **последовательные ﻿7﻿ дней**. Оценки представляют собой последовательность целых чисел от ﻿2﻿ до ﻿5﻿ включительно — по одной оценке на каждый день. Олег хочет выбрать такой непрерывный отрезок своих оценок, чтобы в этом отрезке не было оценок ﻿22 и ﻿3﻿, а количество оценок ﻿5 было максимальным.

Помогите Олегу найти этот особенный момент, когда его школьный свет преобладает над тьмой, и его оценки сияют наиболее ярко!

**Формат входных данных**

Первая строка содержит одно натуральное число ﻿*n*﻿ — количество оценок (1≤ *n* ≤10^5)﻿. Вторая строка содержит ﻿*n*﻿ целых чисел — по оценке ﻿*m*﻿ за каждый день 2≤ *m* ≤5)﻿.

**Формат выходных данных**

Выведите количество пятерок в выбранном Олегом отрезке, удовлетворяющем всем условиям. Если такого отрезка не существует, выведите ﻿−1﻿.

**2 задание**

Ограничение времени 7 секунд

Ограничение памяти 256 МБ

В любом сколько-нибудь приличном редакторе изображений есть функция поворота изображения на ﻿90﻿ градусов. Что уж тут говорить, такая функция есть и в современных мессенджерах! Вот и вам предстоит реализовать эту функцию. Полноценный фоторедактор не потребуется, остановимся только на функции поворота изображения на ﻿90﻿ градусов.

Для простоты будем считать, что изображение представляет из себя матрицу из целых чисел. Вам дана матрица ﻿*n*×*m*﻿. Необходимо вывести матрицу, которая будет являться поворотом исходной на ﻿90﻿ градусов по часовой стрелке.

**Формат входных данных**

Первая строка содержит два натуральных числа ﻿*n*﻿ и ﻿*m*﻿ (1≤ *n*, *m* ≤10^3)﻿. Следующие ﻿*n*﻿ строк содержат описание матрицы, по ﻿*m*﻿ целых неотрицательных чисел, не превосходящих ﻿10^18﻿.

**Формат выходных данных**

Выведите ﻿*m*﻿ строк по ﻿*n*﻿ элементов каждую — повернутую на ﻿90﻿ градусов матрицу.

**3 задание**

Ограничение времени 1 секунда

Ограничение памяти 256 МБ

Понятная файловая система — залог успеха любой операционной системы. К сожалению, не каждая файловая система может похвастаться таким свойством. Но, как говорится, если что-то хочешь сделать хорошо — сделай это сам! Хочется иметь удобное для просмотра представление директорий, чтобы можно было видеть, какие директории в какие вложены.

Для этого требуется по списку директорий вывести их перечисление в алфавитном порядке. При этом вложенные директории должны быть выведены с отступом на два пробела больше, чем у родительской.

**Формат входных данных**

В первой строке дано число ﻿*n*﻿ — количество директорий (1≤ *n* ≤10^5)﻿. В следующих ﻿*n*﻿ строках по одному в строке заданы абсолютные пути ко всем директориям, каждый абсолютный путь — это последовательность вложенных папок, начиная с корневой, разделенная символами "﻿/﻿".

**Гарантируется**, что первая директория во всех путях одинаковая и имеет непустое имя. Имена всех директорий состоят из маленьких латинских букв и имеют длину не более ﻿10﻿. **Гарантируется**, что если директория выведена, то выведены и все, в которые она вложена.

**Формат выходных данных**

Выведите перечисление всех директорий, в котором все директории внутри одной упорядочены по алфавиту, вложенные идут сразу после родительской и имеют отступ на два пробела больше, чем у нее.

**4 задание**

Ограничение времени 3 секунды

Ограничение памяти 4 МБ

В одной из предыдущих задач требовалось вывести перевернутую матрицу, теперь задача усложняется:

При этом поворот необходимо осуществлять **in−place**, т. е. без выделения дополнительной памяти. Для этого вместо результирующей матрицы необходимо вывести **последовательность операций**. За одну операцию можно обменять местами два элемента матрицы.

Вам дана матрица ﻿*n*×*n*﻿, а также указано, надо ли повернуть изображение по часовой ﻿*R*﻿ или против часовой ﻿*L*﻿ стрелки. Выведите последовательность операций, чтобы исходная матрица повернулась на ﻿90﻿ градусов в указанном направлении.

Заметьте, что необязательно переставлять элементы в том порядке, в котором происходил бы поворот, главное, чтобы в результате матрица соответствовала повороту на 90 градусов. Также необязательно, чтобы количество операций было минимальным, нужно только вписаться в ограничения.

**Формат входных данных**

Первая строка содержит одно натуральное число ﻿*n*﻿ (1 ≤*n* ≤10^3)﻿ и указание направления поворота — символ ﻿*R*﻿ или ﻿*L*﻿. Следующие ﻿*n*﻿ строк содержат описание матрицы, по ﻿*n*﻿ целых неотрицательных чисел, не превосходящих ﻿10^18﻿.

**Формат выходных данных**

В первой строке выведите число ﻿*k*﻿ — необходимое количество операций, при этом это число не должно превосходить ﻿7*n^*2﻿. В последующих ﻿*k*﻿ строках выведите по две пары чисел — координаты (*x*1​, *y*1​)﻿ и (*x*2​, *y*2​)﻿ ячеек, между которыми необходимо обменять элементы матрицы.

Замечание

Обратите внимание, что нумерация строк и столбцов матрицы ведётся с ﻿0﻿, а не с ﻿1﻿.

**5 задание**

Ограничение времени 1 секунда

Ограничение памяти 256 МБ

Пошел как-то лесник в лес по грибы, да не в абы какой лес! В клетке либо трава зеленая, либо грибочки белые, либо кусты кусачие. Кусачие кусты, разумеется, непроходимые. Трава зеленая скучная, а грибочки белые, разумеется, по-настоящему интересные.

Лес можно представить в виде клетчатой таблицы размера ﻿*n*×3﻿. Свою дорогу лесник начинает в любой из трех клеток первой строки. После чего каждый раз он может переместиться на следующую строку в соседнюю по углу или стороне клетку, если такая существуют и там не кусты кусачие. Более формально, находясь в клетке (*i*, *j*)﻿ он может переместиться в одну из трех доступных для прохода клеток (*i* + 1, *j* – 1)﻿, (*i* + 1, *j*)﻿ и (*i* + 1, *j* + 1)﻿, если они существуют и там нет кустов.

Леснику, конечно же, интересны грибочки белые, поэтому он хочет знать, какое максимальное их количество он может посетить за прогулку. Если лесник упирается в клетку, из которой никуда не может пойти, он заканчивает свою прогулку.

**Формат входных данных**

В первой строке задано число ﻿*n*﻿ — количество строк в лесу (1≤ *n* ≤10^4)﻿. В следующих ﻿*n*﻿ строках дано по три символа, характеризующих данную строку. Каждый символ равен «﻿.﻿», если в клетке только трава зеленая, «﻿*C*﻿», если в этой клетке растут грибочки белые, и «﻿*W*﻿», если кусты кусачие. Если в первой строке во всех клетках находятся кусты, прогулка лесника заканчивается, не успев начаться.

**Формат выходных данных**

Выведите одно число — наибольшее количество грибов, которые лесник сможет собрать за одну такую прогулку.

**6 задание**

Ограничение времени 1 секунда

Ограничение памяти 256 МБ

Ну и конечно же задача на блуждания коня по шахматной доске размера ﻿*n*×*n*﻿. Чтобы блуждать не было скучно, на доске разбросаны специальные фишки.

Есть два типа фишек — "﻿*K*﻿" и "﻿*G*﻿". При ходе в клетку, в которой лежит фишка "﻿*K*﻿", фигура превращается в коня. При ходе в клетку, в которой лежит фишка "﻿*G*﻿", фигура превращается в короля. Разумеется, после превращения фигура начинает ходить соответственно своему новому типу. Попадание короля в клетку с фишкой "﻿*G*﻿" или коня в клетку с фишкой "﻿*K*﻿" ничего не меняет. При этом трансформация является обязательной, и фигура не может пройти такую клетку с фишкой без превращения в указанный тип.

Ваша задача определить, за какое минимальное количество ходов фигура (возможно в образе коня/короля) доберется до заданной клетки. Заметьте, что количество трансформаций считать не нужно.

**Формат входных данных**

В первой строке задано одно натуральное число ﻿*n*﻿ — размер доски (2≤ *n* ≤ 100)﻿. В следующих ﻿*n*﻿ клетках задано описание шахматной доски — по ﻿*n*﻿ символов. Фишки обозначаются "﻿*K*﻿" и "﻿*G*﻿", а пустые клетки за "﻿.". Начальная клетка обозначается "﻿*S*﻿", а конечная — "﻿*F*﻿".

Гарантируется, что на начальной и конечной клетках нет фишки.

**Формат выходных данных**

Выведите единственное число — необходимое количество ходов. Если такого пути не существует, то выведите ﻿−1﻿.

**Замечание**

Как и всегда, конь ходит буквой ﻿Г﻿, т.е. на одну клетку в одну сторону и две клетки в другую, всего до ﻿8﻿ возможных ходов. Король может перейти из текущей клетки в соседнюю по стороне или углу, всего до ﻿8﻿ возможных ходов.