Лимонад

Известный шеф-повар готовит свой лимонад с заботой о своих взыскательных клиентах. Он понимает, что важным рецептом лимонада является выбор хороших, неиспорченных лимонов. Но эта задача сложнее, чем кажется на первый взгляд, потому что со временем один лимон может быть испорчен другим и, в свою очередь, может испортить другие неиспорченные соседние лимоны.

Чтобы гарантировать, что все лимоны, выбранные для лимонада, хорошие, он думает, что ему нужна помощь компьютерного ученого для создания программы. Программа должна помочь выбрать лучшие лимоны, избегая испорченных или потенциально испорченных лимонов. Только так шеф-повар может гарантировать своим клиентам идеальный лимонад. Увлекаясь информатикой, он предполагает, что ящик с лимонами можно представить в виде двумерной таблицы, пометив хорошие лимоны цифрой 1 (один), испорченные — заглавной латинской буквой X, пустое место в ящике — 0 (нулем). Наблюдения шефа показывают, что измененный лимон может воздействовать на соседние лимоны только в четырех направлениях клетки – вверх, вниз, влево, вправо.

Задание. Разработать программу, которая бы анализировала входную двухмерную таблицу, содержавшую распределение хороших и плохих лимонов, и сгенерировала двухмерную таблицу с лимонами, которые со временем могут испортиться (отмечая их заглавной буквой X), чтобы исключить их из приготовления лимонада.

Входные данные. Файл `limonada.in` содержит размер массива N и M, где N — количество строк, а M — количество столбцов. Следующие N строк в файле содержат M значений, каждое из 1, 0 или X (не разделенных пробелами или другими символами).

Выходные данные. Файл `limonada.out` содержит двумерную таблицу с лимонами, которые могут испортиться (они будут отмечены заглавной латинской буквой X) и лимонами, которые останутся годными спустя длительное время (они будут отмечены цифрой 1).

Ограничения. Максимальное значение для N и M равно 100. Время выполнения не должно превышать `1` секунду. Программа должна использовать не более `8` мегабайт оперативной памяти. Исходный файл должен называться `limonada.pas`, `limonada.c` или `limonada.cpp`.

Пример.

limonada.in

```
4 5
10001
01111
10X00
10011
```

limonada.out

```
1000X
0XXXX
10X00
10011
```

limonada.in

```
5 6
000010
001101
01X101
100001
0X1101
```

limonada.out

```
000010
00XX01
0XXX01
100001
0XXX01
```

Объяснение. В первом примере исходная матрица содержит только один плохой лимон в ячейке (3, 3). Лимон, по наблюдениям, может испортить в первую очередь только соседа по ячейке (2,3) – направление вверх. Этот лимон в свою очередь портит лимоны в ячейках (2,2) — левое направление, и (2, 4) — правое направление. Затем справа от клетки (2, 4) есть лимон, который тоже испортится, ячейка (2,5). А последний испортит лимон в клетке (1,5). Поэтому повар сможет использовать только пять лимонов из ячеек (1,1), (3, 1), (4, 1), (4,4) и (4,5) — у них нет соседей, которые могут испортить их, по направлениям - *вверх, вниз, влево, вправо*.

Второй пример начинается с двух гнилых лимонов в ячейках (3,3) и (5,2). Ячейки (3,2), (3,4) и (2,3) испортятся от ячейки (3, 3); по направлениям - влево, вправо и вверх, соответственно. А ячейка (2, 4) испортится от ячейки (2,3), испорченной ранее. Клетки (5,3) и (5,4) будут испорчены последовательно, в правом направлении от лимона клетки (5,2). В этом случае шеф сможет использовать в лимонаде только 6 лимонов.