

Codeforces Round #470 (Div. 2)

Problem A – Protect Sheep

Prof. Edson Alves

Faculdade UnB Gama

Bob is a farmer. He has a large pasture with many sheep. Recently, he has lost some of them due to wolf attacks. He thus decided to place some shepherd dogs in such a way that all his sheep are protected.

The pasture is a rectangle consisting of $R \times C$ cells. Each cell is either empty, contains a sheep, a wolf or a dog. Sheep and dogs always stay in place, but wolves can roam freely around the pasture, by repeatedly moving to the left, right, up or down to a neighboring cell. When a wolf enters a cell with a sheep, it consumes it. However, no wolf can enter a cell with a dog.

Initially there are no dogs. Place dogs onto the pasture in such a way that no wolf can reach any sheep, or determine that it is impossible. Note that since you have many dogs, you do **not** need to minimize their number.

Bob é um fazendeiro. Ele tem um pasto largo com muitas ovelhas. Recentemente, ele perdeu algumas delas devido a ataques de lobos. Assim ele decidiu colocar alguns cães pastores de modo que todas suas ovelhas estejam protegidas.

O pasto é um retângulo composto por $R \times C$ células. Cada célula ou está vazia, ou contém uma ovelha, ou um lobo ou um cão. Ovelhas e cães sempre ficam no lugar, mas lobos podem andar livremente pelo pasto, movendo-se para as células vizinhas à esquerda, à direita, acima ou abaixo. Quando um lobo entra em um célula contendo uma ovelha, ele a devora. Contudo, nenhum lobo pode entrar em uma célula contendo um cão.

Inicialmente não há cães. Posicione cães no pasto de tal modo que nenhum lobo possa alcançar nenhuma ovelha, ou determine que esta tarefa é impossível. Note que, uma vez que você tem muitos cães, você **não** precisa minimizar o seu número.

Input

First line contains two integers R ($1 \leq R \leq 500$) and C ($1 \leq C \leq 500$), denoting the number of rows and the numbers of columns respectively.

Each of the following R lines is a string consisting of exactly C characters, representing one row of the pasture. Here, 'S' means a sheep, 'W' a wolf and '.' an empty cell.

Entrada

A primeira linha contém dois inteiros R ($1 \leq R \leq 500$) e C ($1 \leq C \leq 500$), denotando o número de linhas e o número de colunas, respectivamente.

Cada uma das R linhas seguintes contém um string formada por exatamente C caracteres, representando uma linha do pasto. Aqui, 'S' significa ovelha, 'W' um lobo e '.' uma célula vazia.

Output

If it is impossible to protect all sheep, output a single line with the word “No”.

Otherwise, output a line with the word “Yes”. Then print R lines, representing the pasture after placing dogs. Again, 'S' means a sheep, 'W' a wolf, 'D' is a dog and ' ' an empty space. You are not allowed to move, remove or add a sheep or a wolf.

If there are multiple solutions, you may print any of them. You don't have to minimize the number of dogs.

Saída

Se é impossível proteger todas as ovelhas, imprima uma única linha com a palavra “No”.

Caso contrário, imprima uma linha com a palavra “Yes”. Então imprima R linhas, representando o pasto após o posicionamento dos cães. Novamente, 'S' significa uma ovelha, 'W' um lobo, 'D' é um cão e '.' um espaço vazio. Não é permitido mover, remover ou adicionar uma ovelha ou um lobo.

Se há múltiplas soluções, imprima qualquer uma delas. Você não precisa minimizar o número de cães.

Exemplo de entrada e saída

Exemplo de entrada e saída

6 6

Exemplo de entrada e saída

6 6 ← # de colunas
↑
de linhas

Exemplo de entrada e saída

6 6

..S...

..S.W.

.S....

..W...

...W..

.....

Exemplo de entrada e saída

```
6 6
..S...
..S.W.
.S....
..W...
...W..
.....
```

Possibilidade: Construir um “muro” de cães

Exemplo de entrada e saída

6 6
..SD..
..SDW..
.SD...
DDW...
...W..
.....

Possibilidade: Construir um “muro” de cães

Exemplo de entrada e saída

1 2

SW

Exemplo de entrada e saída

1 2

SW

Impossível: Um lobo é vizinho de uma ovelha!

Solução

6 6
..S...
..S.W.
.S....
..W...
...W..
.....

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W..
.....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W...
.....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W...
....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W..
.....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W..
.....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

```
6 6
..S...
..S.W.
.S....
..W...
...W..
.....
```

O alcance de um lobo pode ser determinado por uma BFS

Solução

6 6
..S...
..S.W.
.S....
..W...
...W..
.....

Solução

6 6

..S...

..S.W.

.S....

..W...

...W..

.....

Contudo, cães são ilimitados!

Solução

6 6

..S...

..S.W.

.S....

..W...

...W..

.....

Contudo, cães são ilimitados!

Se nenhum lobo é vizinho de ovelha, coloque cães à vontade!

Solução

6 6

DDSDDD

DDSDWD

DSDDDD

DDWDDD

DDDWDD

DDDDDD

Contudo, cães são ilimitados!

Se nenhum lobo é vizinho de ovelha, coloque cães à vontade!

```

bool solve(int R, int C) {
    for (int r = 1; r <= R; ++r) {
        for (int c = 1; c <= C; ++c) {
            if (M[r][c] != 'S')
                continue;

            vector<ii> ps { {r-1, c}, {r+1, c}, {r, c-1}, {r, c+1} };

            for (auto [x, y] : ps) {
                if (M[x][y] == 'W')
                    return false;

                if (M[x][y] == '.')
                    M[x][y] = 'D';
            }
        }
    }

    return true;
}

```