Screamers in the Storm

`screamers.c`, `screamers.cpp`, `Screamers.java`, `screamers.py`

As you might remember from your first years in school, the human race invented beer brewing at least about 7000 years ago. The total amount of beer consumed from those times must be monumental and surely the rate of consumption is not going to shrink in the coming millennia.

To celebrate these facts, we invite you to implement a game, seemingly unrelated to beer brewing. It is quite possible, however, that after a successful implementation you might feel a little dizzy, just as if you have had a little bit more than your daily dose of beer...

The game is played on a rectangular $M \times N$ grid consisting of square tiles of different types.

Animals and food

There is some number of animals on the grid, each animal occupies exactly one tile.

Grass grows, sheep eat grass, wolves eat sheep, and both wolves and sheep can die of starvation.

Turns and animal actions

Each turn consists of actions in the following order:

1. All animals on the grid move. Each wolf moves to a neighbouring tile in the east (to the right). If the wolf's move is not possible, the wolf moves to the westernmost tile in its row. Each sheep moves to a neighbouring tile in the south (down). If the sheep's move is not possible, the sheep moves to the northernmost tile in its column.
2. If a wolf and a sheep occupy the same tile, the wolf eats the sheep and the tile is changed to a *Soil with carcass* tile.
3. If a sheep is located on a *Soil with grass* tile, the sheep eats the grass and the tile is changed to a *Soil* tile.
4. If a wolf hasn't eaten in any of the last 10 turns including the current turn, it dies and the tile is changed to a *Soil with carcass* tile.
5. If a sheep didn't eat in any of the last 5 turns including the current turn, it dies and the tile is changed to a *Soil with carcass* tile.

Types of tiles and their changes

There are three types of tiles. The type of a tile may be changed in the course of the game.

1. *Soil* tile: After 3 turns after the beginning of the game or after 3 turns after the tile became a *Soil* tile, the tile becomes a *Soil with grass* tile.
2. *Soil with grass* tile: If the grass is eaten by a sheep, the tile immediately becomes a *Soil* tile. The grass will grow again at the tile after 3 turns.

3. *Soil with carcass* tile: Whenever an animal dies on a tile of any type, the tile immediately becomes a *Soil with carcass* tile. Animals can still move to this tile, but the grass will never grow on this tile again. As the game progresses, more carcasses may accumulate on the tile.

Input Specification

First line of the input contains three integers T, N and M ($1 \leq T \leq 100$, $1 \leq M, N \leq 20$), where T is the number of turns, M is the number of rows and N is the number of columns of the grid. The following M lines contain N characters each. Characters denote the types of tiles:

- `.` (dot character) denotes a *Soil* tile
- `S` denotes a *Soil* tile with a sheep on it
- `W` denotes a *Soil* tile with a wolf on it

Output Specification

Output M lines each containing N characters describing the state of the grid after the end of the T-th turn. If there is an animal on a tile, output:

- `W` for a tile with a wolf on it
- `S` for a tile with a sheep on it

Otherwise, output:

- `*` for a *Soil with carcass* tile
- `#` for a *Soil with grass* tile
- `.` (dot character) for a *Soil* tile

Sample Input 1

```
6 6 5
..S..
.....
.S...
.....
....W
.S...
```

Output for Sample Input 1

```
##S##
#####
#####
#.###
W*.##
#S.##
```

Sample Input 2

```
14 3 3
S..
W..
...
```

Output for Sample Input 2

```
.##
#*#
S##
```

Sample Input 3

```
2 3 1
S
.
.
```

Output for Sample Input 3

```
.
.
S
```

Sample Input 4

```
3 3 1
S
.
.
```

Output for Sample Input 4

```
S
#
#
```

Sample Input 5

```
4 3 1
S
.
.
```

Output for Sample Input 5

```
#
S
#
```

Sample Input 6

```
5 3 1
S
.
.
```

Output for Sample Input 6

```
#
.
S
```