

---

## Velociraptors 101

P72222\_en

Olimpiada Informática Española — Final 2007 (2007)

---

You are on the point of arriving to the ground after your first jump in parachute, when you discover that somebody has released a group of velociraptors in the runway. Oh, well! This kind of things happen sometimes. Now, all of them are quiet, but as soon as you arrive to the ground, you know they will attack you following the shortest way (they are smart), and you will remain immobile, gripped by fear and the material of the parachute.

The runway is rectangular. We mark with a dot the empty spaces, with a 'v' the velociraptors, and with a '#' the squares with obstacles (you cannot land there, and velociraptors cannot cross them). (Observe the test data to get an idea). Velociraptors, that take a second to cover a square, can move horizontal and vertically, but not in diagonal. You are asked to mark with a 'x' the free squares where, landing in them, you will maximize your (brief but intense) remaining time life.

### Input

A test data contains various cases. Each case starts with a line with two numbers  $R$  and  $C$  (dimensions in rows and columns of the runway map). Then, there are  $R$  rows of  $C$  caracteres each one, with the description of the map as it has been previously explained.

We assure you that an empty square to land will always exist, therefore, your life time will always be greater than 0, and there is not any closed space without velocitaptors, that is, that for any empty square always exists a velociraptor that can reach it, so your life time will not be infinite.

### Output

For each case, your program must print the map where you will mark with a 'x' the squares that, landing in them, you will maximize your time life. Separate two maps by a line with 3 dashes '---', as in the instance.

### Scoring

- **Test1:** 100 cases where  $2 \leq F, C \leq 10$ .

25 Points

- **Test1:** 50 cases where  $2 \leq F, C \leq 100$ .

40 Points

- **Test1:** 10 cases where  $2 \leq F, C \leq 500$ .

35 Points

### Sample input 1

```
6 6
.....
.....
.V....
.....
.....
.....V
```

```
3 10
...#.....
...#...#.V.
.V....#...
```

---

```
3 10
V###...V..
...#V.#.V.
```

.V.V.V#...

3 10

V###.V.V.V  
..##V.#.V.  
.V.V.V##.V

### Sample input 2

6 6

VVVVVV  
V...V  
V...V  
V...V  
V...V  
VVVVVV

6 5

VVVVV  
V...V  
V...V  
V...V  
V...V  
VVVVV

5 6

.....  
.....  
...V..  
.....  
.....

5 6

VVVVVV  
VVVV.V  
V.VV.V  
V.##.V  
VVVVVV

5 6

VVVVVV  
VVV.VV  
VV...V  
VVV.VV  
VVVVVV

### Sample output 1

....XX

.....

.V....

.....

.....

.....V

---

...#X.....

...#.X#.V.

.V....#...

---

V###.X.V.X

.X#V.#.V.

.V.V.V#X.X

---

V###XVXVXV

XX##VX#XVX

XVXVXV##XV

### Sample output 2

VVVVVV

V...V

V.XX.V

V.XX.V

V...V

VVVVVV

---

VVVVV

V...V

V.X.V

V.X.V

V...V

VVVVV

---

X.....

.....

...V..

.....

X.....

---

VVVVVV

VVVVXV

VXVVXV

VX##XV

VVVVVV

---

VVVVVV

VVV.VV

VV.X.V

VVV.VV

VVVVVV

