

Sheryl works for a software company in the country of Brada. Her job is to develop a Windows operating system. People in Brada are incredibly conservative. They even never use graphical monitors! So Sheryl's operating system has to run in text mode and windows in that system are formed by characters. Sheryl decides that every window has an ID which is a capital English letter ('A' to 'Z'). Because every window had a unique ID, there can't be more than 26 windows at the same time. And as you know, all windows are rectangular.

On the screen of that ugly Windows system, a window's frame is formed by its ID letters. Fig-1 shows that there is only one window on the screen, and that window's ID is 'A'. Windows may overlap. Fig-2 shows the situation that window B is on the top of window A. And Fig-3 gives a more complicated overlapping. Of course, if some parts of a window are covered by other windows, you can't see those parts on the screen.

```

.....
...AAAAAAAAAAAAA.....
...A.....A.....
...A.....A.....
...A.....A.....
...AAAAAAAAAAAAA.....
.....

```

Fig-1

```

.....
...AAAAAAAAAAAAA.....
...A.....A.....
...A.....BBBBBBBBB...
...A.....B.....B...
...AAAAAAAAB.....B...
...BBBBBBBBBB...
.....

```

Fig-2

```

.....
...AAAAAAAAAAAAA.....
...A.....A.....
...A.....BBBBBBBBB...
...A.....B.....BCCC.
...AAAAAAAAB.....B..C.
...C...BBBBBBBBBB..C.
...CCCCCCCCCCCCCCCC.
.....

```

Fig-3

If a window has no parts covered by other windows, we call it a "top window" (The frame is also considered as a part of a window). Usually, the top windows are the windows that interact with user most frequently. Assigning top windows more CPU time and higher priority will result in better user experiences. Given the screen presented as Figs above, can you tell Sheryl which windows are top windows?

### Input

The input contains several test cases.

Each test case begins with two integers,  $n$  and  $m$  ( $1 \leq n, m \leq 100$ ), indicating that the screen has  $n$  lines, and each line consists of  $m$  characters.

The following  $n$  lines describe the whole screen you see. Each line contains  $m$  characters. For characters which are not on any window frame, we just replace them with '.'.

The input ends with a line of two zeros.

It is guaranteed that:

- 1) There is at least one window on the screen.
- 2) Any window's frame is at least 3 characters wide and 3 characters high.
- 3) No part of any window is outside the screen.

### Output

For each test case, output the IDs of all top windows in a line without blanks and in alphabet order.

### Sample Input

```

9 26
.....
...AAAAAAAAAAAAA.....
...A.....A.....
...A.....BBBBBBBBB...
...A.....B.....BCCC.
...AAAAAAAAB.....B..C.
...C...BBBBBBBBBB..C.
...CCCCCCCCCCCCCCCC.
.....
7 25
.....
...DDDDDDDDDDDD.....
...D.....D.....
...D.....D.....
...D.....D...AAA...
...DDDDDDDDDDDD..A.A...
...AAA...
0 0

```

### Sample Output

```

B
AD

```