

# Problem F

## Finding Seats

*Source file name: find.c, find.cpp or find.java*

A group of  $K$  friends is going to see a movie. However, they are too late to get good tickets, so they are looking for a good way to sit all nearby. Since they are all science students, they decided to come up with an optimization problem instead of going on with informal arguments to decide which tickets to buy.

The movie theater has  $R$  rows of  $C$  seats each, and they can see a map with the currently available seats marked. They decided that seating close to each other is all that matters, even if that means seating in the front row where the screen is so big it's impossible to see it all at once. In order to have a formal criteria, they thought they would buy seats in order to minimize the *extension* of their group.

The *extension* is defined as the area of the smallest rectangle with sides parallel to the seats that contains all bought seats. The area of a rectangle is the number of seats contained in it. They've taken out a laptop and pointed at you to help them find those desired seats.

### Input

Each test case will consist on several lines. The first line will contain three positive integers  $R$ ,  $C$  and  $K$  as explained above ( $1 \leq R, C \leq 300$ ,  $1 \leq K \leq R \times C$ ). The next  $R$  lines will contain exactly  $C$  characters each. The  $j$ -th character of the  $i$ -th line will be 'X' if the  $j$ -th seat on the  $i$ -th row is taken or '.' if it is available. There will always be at least  $K$  available seats in total.

Input is terminated with  $R = C = K = 0$ .

*The input must be read from standard input.*

### Output

For each test case, output a single line containing the minimum extension the group can have.

*The output must be written to standard output.*

<b>Sample input</b>	<b>Output for the sample input</b>
3 5 5 ...XX .X.XX XX... 5 6 6 ..X.X. .XXX.. .XX.X. .XXX.X .XX.XX 0 0 0	6 9