ACM International Collegiate Programming Contest

Western Europe Regional Contest

’s-Hertogenbosch, 13–14 November 1999



Problem E

Papergirl Paradise

Input: e.in

The life of a papergirl is not always easy. Suburbs often have a maze of little streets, and subscribers to the newspaper may live quite far apart. Compared to that, a papergirl who makes her round in the city center of Skyscraper Sity has a far simpler task. There are often enough subscribers in one single skyscraper from which to make a living, so the round of a papergirl rarely consists of more than one flat. A papergirl just starts at the ground floor of the flat and works her way up to the top, delivering the newspaper on each floor as she goes along. The only thing she has to take care of is that each subscriber gets her/his newspaper before 8:00 in the morning.

Jill is one of the papergirls in Skyscraper Sity, and she is given a number of skyscrapers to choose from for her round. Since she wants to earn money, but doesn’t want to get up too early in the morning, she tries to figure out which skyscraper takes the least amount of time in which to deliver the newspapers. For each skyscraper she knows the layout of the skyscraper, including the position of the entrance on the ground floor and where each of the subscribers is located. She knows that in each flat the stairs are always at the left end and the right end of the flat, and they always go up from the ground floor all the way to the top. She has set one ground rule for herself: she doesn’t want to take more stairs than strictly necessary with all those newspapers on her back, so she will always deliver all the newspapers on a floor before moving on to the next floor.

You are asked by Jill to help her determine the time it would take for her to deliver all the newspapers in a given skyscraper.

Input

The first line of the input contains the number of skyscrapers S ($1 = S = 100$). For each skyscraper there follows one line with 2 integers: f ($1 ? f ? 30$) and w ($4 ? w ? 80$), giving the number of floors and the width of each floor in the skyscraper. Then exactly $f+1$ lines will follow giving the layout of the skyscraper, where every line will contain exactly w characters.

The first line is the roof of the skyscraper, which is a ‘+’, followed by $w-2$ ‘-’ characters and a ‘+’ again.

Each of the following lines starts and ends with a ‘%’ (representing the stairs at the left and right side of the flat), and in between are by $w-2$ characters from the following set:

*	The apartment of a subscriber to the newspaper
.	The apartment of someone who is not a subscriber to the newspaper, and therefor irrelevant for you.

On the ground floor (that is the last floor in the input of a skyscraper), there is also exactly one ‘@’ character between the stair symbols, which denotes the entrance to the skyscraper. The number of subscribers on a floor is only limited by the width of the floor. There will always be at least one subscriber on the top floor.

A newspaper is delivered by moving past a subscriber apartment. Moving up a stair takes the same amount of time as moving one step in a horizontal direction.

Take a look at the next example input, with 5 floors (*f*=5) and 10 apartments per floor (*w*=12):

```
5 12
+----------+
%...*..*..*%      Fourth floor = top floor
%.........%       Third floor
%*..*******%      Second floor
%*.*.***.*.%      First floor
%...@.*...*%      Ground floor
```

[Note: the names of the floors are not in the real input!]

In this example the entrance is at position 3 on the ground floor. There are 2 subscribers on the ground floor (at locations 5 and 9), 6 at the first floor, 8 at the second floor but none at the third floor, while there are 3 subscribers at the top floor.

Output

For each test case, you must output a line containing the least number of steps it takes to deliver the last newspaper.

Sample Input	Output for the Sample Input
2 5 12 +----------+ %...*..*..*% %.........% %*..*******% %*.*.***.*.% %...@.*...*% 10 10 +--------+ %...**...% %**.....*% %.*....**% %.**...*.% %..*..**.% %..**.*..% %...***..% %...**...% %....*..*% %@......*%	40 93