Rush to Cornetti (`cornetti`)

Antonio and William are two friends from Italy, and after a lively night out at a local party with their foreign friends, they decide to share a beloved Italian tradition: getting warm *cornetti* at dawn from the most famous bar in the city. It's one of those unforgettable experiences, and they're eager to show their friends the joy of savoring a fresh pastry after a long night out. This tradition is particularly strong in Antonio's hometown of Catania, where young people often gather after a night of fun to indulge in a late-night treat at one of the popular local spots.

The night is chilly, and everyone is starving, so they need to find the quickest route from the party venue (**P**) to the bar (**B**). The map is represented as a rectangular grid of size $N \times M$. However, the streets of the city are not so straightforward - some paths are blocked by old buildings (#), and others are narrow alleys (.) where only foot traffic is allowed. Antonio and William know the city like the back of their hands, but with some shortcuts involving hidden shuttle stops (**S**) that can make the journey quicker, they need to carefully plan the best route to get their friends to the destination in the least amount of time.

Figure 1: Freshly baked cornetti.

Help Antonio and William find the fastest way to reach the bar so that they can enjoy their cornetti before sunrise, with movement allowed only in the four cardinal directions (up, down, left, right) and not diagonally. Every movement between two adjacent cells takes exactly 1 minute on foot. If they reach a shuttle stop, they can use the shuttle to travel to any other stop in the city in just 1 minute (regardless of the distance).

They want to make sure their foreign friends have the perfect Italian experience, but the hunger is real, and time is ticking!

Input

The first line contains two integers: N and M, respectively the number of rows and columns of the map.

Each of the following N lines contains a string of M characters describing the map. Each character C_{ij} corresponds to a specific part of the city: '**P**' represents the party venue, '**B**' represents the cornetti bar, '**.**' represents walkable streets, '#' represents impassable areas, and '**S**' represents shuttle stops.

Output

You need to write a single line containing a single integer, the minimum number of moves required to reach **B** from **P**. If there is not a valid path from **P** to **B**, return -1.

Constraints

- $2 \leq N, M \leq 1000$.
- The grid contains exactly one **B** and exactly one **P**.

Examples

input	output
5 4 P.#. #... S... .... .S.B	7
7 7 S.....S ....B.. ....P.... .... S.....S	4
5 3 P.# #.. .S. ..# ..B	6
3 3 P.. ..# .#B	-1

Explanation

In the **first sample case** one possible optimal solution is for Antonio and William to reach a shuttle stop in 4 minutes, use it to travel to another shuttle stop in 1 minute, and finally walk for 2 more minutes to their destination.

In the **second sample case** the city is big but the party venue **P** and the bar **B** are pretty close to each other, so Antonio and William can simply walk for 4 minutes.

In the **third sample case** it's not necessary to use the shuttle, they can simply walk for 6 minutes and that's already the best that can be done.

In the **fourth sample case** there is no way to reach the destination.