

VUK

Vjekoslav the Wolf is running away from a bunch of blood hungry hunters. The hunters are smart and hide behind trees. Vjekoslav knows this, but doesn't know which trees. He would like to run to his comfortable, civilized cottage (as opposed to the hunters quite uncivilized den, yes I am rooting for the Wolf here) staying as far away as possible from any trees.

The forest can be represented as a N by M grid. Let us mark empty meadow patches with '.', patches with a tree in the middle with '+', Vjekoslav's current position with 'V' and the position of his cottage with 'J'. Vjekoslav can run from his current patch to any other patch north, east, south or west from him, even if it contains a tree.

If Vjekoslav is standing in R -th row and C -th column on the grid and there is a tree in the A -th row and B -th column then the distance between Vjekoslav and that tree is:

$$|R-A| + |C-B|$$

Help Vjekoslav find the best route to his cottage. The best route is any route that maximizes the minimal distance between Vjekoslav and all trees at any given moment. Note that Vjekoslav's cottage doesn't occupy the entire patch so that patch must also be included in the route.

Input

The first line of input contains integers N and M ($1 \leq N, M \leq 500$), grid dimensions.

The next N lines contain M characters each: '.', '+', 'V', 'J'.

Input will contain exactly one character 'V' and 'J' and at least one character '+'.

Output

Output a single integer, the minimal distance from a tree in the optimal route.

Example

Input:

4 4

+...

....

....

V..J

Output:

3