

# Store-keeper

The floor of a store is a rectangle divided into  $n*m$  square fields. Two fields are adjacent, if they have a common side. A parcel lays on one of the fields. Each of the remaining fields is either empty, or occupied by a case, which is too heavy to be moved by a store-keeper. The store-keeper has to shift the parcel from the starting field **P** to the final field **K**. He can move on the empty fields, going from the field on which he stands to a chosen adjacent field. When the store-keeper stays on a field adjacent to the one with the parcel he may push the parcel so that it moves to the next field (i.e. the field on the other side of the parcel), assuming condition that there are no cases on this field.

## Task

Write a program, which:

- reads from the standard input a store scheme, a starting position of the store-keeper and a final position of the parcel,
- computes minimal number of the parcel shifts through borders of fields, which are necessary to put the parcel in the final position or decides that it is impossible to put the parcel there,
- writes the result into the standard output.

## Input

The number of test cases  $t$  is in the first line of input, then  $t$  test cases follow separated by an empty line. In the first line of each test case two positive integers separated by a single space,  $n, m \leq 100$ , are written. These are dimensions of the store. In each of the following  $n$  lines there appears one  $m$ -letter word made of letters S, M, P, K, w. A letter on  $i$ -th position in  $j$ -th word denotes a type of the field with coordinates  $(i, j)$  and its meaning is following:

- S - case,
- M - starting position of the store-keeper,
- P - starting position of the parcel,
- K - final position of the parcel,
- w - empty field.

Each letter M, P and K appears in the test case exactly once.

## Output

Your program should write to the standard output for each test case:

- exactly one word NO if the parcel cannot be put on the target field,
- exactly one integer, equal to the minimal number of the parcel shifts through borders of the fields, necessary to put a parcel on a final position, if it is possible to put the parcel there.

## Example

Sample input:

1  
10 12  
SSSSSSSSSSSS  
SwwwwwwSSSS  
SwSSSSwwSSSS  
SwSSSSwwSKSS  
SwSSSSwwSwSS  
SwwwwwPwwwww  
SSSSSSSwSwSw  
SSSSSSMwSwww  
SSSSSSSSSSSS  
SSSSSSSSSSSS

**Sample output**

7