***Coffe-Break***

***Autor: Júnior Andrade***

***Time Limit: 2s***

Os estudantes de programação ficam extremamente famintos antes do aquecimento, uma vez que acabaram de chegar do hotel e estão muito ansiosos para competir / comer. Ao chegar na final mundial, os times brasileiros chegando no hall onde fica o coffe break, se depararam com uma quantidade enorme de times de todas as nacionalidades possiveis. E, como chegaram um pouco atrasados, a comida estava no fim. Para não perderem o coffe-break, os times desejam chegar o quanto antes nas mesas onde essas comidas se encontravam.

O hall onde o coffe-break está sendo realizado é representado por um grid NxM onde N é o numero de linhas e M o numero de colunas desse grid. Cada célula desse grid pode conter um caracter ‘T’ representando um time brasileiro no hall, um caracter ‘#’ representando uma mesa ou um numero [0..9], que representa o tempo necessario para se atravessar essa célula.

Cada mesa pode suportar um e somente um time. Note também que não é possivel “atravessar” por cima de uma mesa para cortar caminho. Porém, é permitido que dois ou mais times estejam na mesma célula. Considere que o tempo necessário para atravessar uma célula que contem um time brasileiro INICIALMENTE posicionado é 0 (pois os times brasileiros se ajudam a acelerar o processo como um todo). Deseja-se que todos os times encontrem uma mesa, e que o tempo que o time que demore mais seja o minimo possivel.

Você pode ajudar a organização a alocar os times de maneira a minimizar a restrição desejada?

**Input**

A entrada contem uma linha contendo dois inteiros N e M (1 <= N, M <= 100) como o descrito acima. Após isso existiram N linhas com M colunas contendo os caracteres {‘#’, ‘T’, ‘[0 ... 9]’} que correspondem a uma mesa, um time, ou a quantidade de tempo gasta para se atravessar uma célula do grid. Haverão no máximo 100 times e 100 mesas.

**Output**

Imprima um inteiro contendo o tempo minimo para o time que demore mais a encontrar uma mesa precise utilizar. Caso não haja resposta, imprima -1.

|  |  |
| --- | --- |
| Input | Output |
| 4 4  11T1  11T1  11#1  222# | 2 |

Neste caso, o T mais acima pode encontrar a mesa mais abaixo, enquanto o T mais abaixo pode simplesmente desfrutar da mesa que está ao seu lado. 2 + 0 = 3. O primeiro time consegue um custo de dois, pois pode utilizar a posição que o time brasileiro mais abaixo estava, e dessa maneira, pagar somente 0 + 1 + 1.

|  |  |
| --- | --- |
| Input | Output |
| 2 4  T1##  TT#1 | -1 |

A resposta é -1, pois um dos times não consegue alcançar a mesa mais a direita.