D. Minesweeper 1D

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Game "Minesweeper 1D" is played on a line of squares, the line's height is 1 square, the line's width is n squares. Some of the squares contain bombs. If a square doesn't contain a bomb, then it contains a number from 0 to 2 — the total number of bombs in adjacent squares.

For example, the correct field to play looks like that: `001*2***101*`. The cells that are marked with "`*`" contain bombs. Note that on the correct field the numbers represent the number of bombs in adjacent cells. For example, field `2*` is not correct, because cell with value 2 must have two adjacent cells with bombs.

Valera wants to make a correct field to play "Minesweeper 1D". He has already painted a squared field with width of n cells, put several bombs on the field and wrote numbers into some cells. Now he wonders how many ways to fill the remaining cells with bombs and numbers are there if we should get a correct field in the end.

Input

The first line contains sequence of characters without spaces $s_1s_2... s_n$ $(1 \le n \le 10^6)$, containing only characters "`*`", "`?`" and digits "`0`", "`1`" or "`2`". If character s_i equals "`*`", then the i-th cell of the field contains a bomb. If character s_i equals "`?`", then Valera hasn't yet decided what to put in the i-th cell. Character s_i, that is equal to a digit, represents the digit written in the i-th square.

Output

Print a single integer — the number of ways Valera can fill the empty cells and get a correct field.

As the answer can be rather large, print it modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
?01???
```

output

```
4
```

input

```
?
```

output

```
2
```

input

```
**12
```

output

```
0
```

input

```
1
```

output

```
0
```

Note

In the first test sample you can get the following correct fields: `001**1`, `001***`, `001*2*`, `001*10`.