

# Expressões

Pedrinho e Zezinho estão precisando estudar resolução de expressões matemáticas para uma prova que irão fazer. Para isso, eles querem resolver muitos exercícios antes da prova. Como sabem programar, então decidiram fazer um gerador de expressões matemáticas.

O gerador de expressões que eles criaram funciona em duas fases. Na primeira fase é gerada uma cadeia de caracteres que contém apenas os caracteres '{', '[', '(', '}', ']' e ')'. Na segunda fase, o gerador adiciona os números e operadores na estrutura criada na primeira fase. Uma cadeia de caracteres é dita *bem definida* (ou válida) se atende as seguintes propriedades:

1. Ela é uma cadeia de caracteres vazia (não contém nenhum caractere).
2. Ela é formada por uma cadeia *bem definida* envolvida por parênteses, colchetes ou chaves. Portanto, se a cadeia  $S$  é *bem definida*, então as cadeias  $(S)$ ,  $[S]$  e  $\{S\}$  também são *bem definidas*.
3. Ela é formada pela concatenação de duas cadeias *bem definidas*. Logo, se as cadeias  $X$  e  $Y$  são *bem definidas*, a cadeia  $XY$  é *bem definida*.

Depois que Pedrinho e Zezinho geraram algumas expressões matemáticas, eles perceberam que havia algum erro na primeira fase do gerador. Algumas cadeias não eram *bem definidas*. Eles querem começar a resolver as expressões o mais rápido possível, e sabendo que você é um ótimo programador (e participa da OBI) resolveram pedir que escreva um programa que dadas várias cadeias geradas na primeira fase, determine quais delas são *bem definidas* e quais não são.

## Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias. Em seguida temos  $T$  linhas, cada uma com uma cadeia  $A$ .

## Saída

Para cada instância imprima uma linha contendo a letra S se a cadeia é *bem definida*, ou a letra N caso contrário.

## Restrições

- $1 \leq T \leq 20$ .
- a cadeia de caracteres  $A$  tem entre 1 e 100000 caracteres.
- a cadeia de caracteres  $A$  contém apenas caracteres '{', '[', '(', '}', ']' e ')

## Exemplos

### Entrada

```
12
()
```

[]  
{  
(  
X  
({  
{()  
{}  
)  
{  
(  
(({{}())){}  
((((((((({}))))))))))

**Saída**

S  
S  
S  
N  
N  
S  
S  
N  
N  
N  
S  
N