

## Casamento de delimitadores

Desenvolva um algoritmo para verificar o uso dos delimitadores de parênteses, colchetes e chaves. A solução deve envolver o uso de uma pilha. A lógica da solução é a seguinte:

- Analise todos os caracteres da expressão
- Quando encontrar um delimitador de abertura ( , [ ou { empilhe-o na pilha.
- Quando encontrar um delimitador de fechamento ) , ] ou }, desempilhe o elemento do topo da pilha e compare com esse delimitador de fechamento. Se forem compatíveis, prossiga com a análise, caso contrário, acuse o erro.

Possíveis erros:

- a) Incompatibilidade de delimitadores de abertura e fechamento. Por exemplo: ( ]
- b) Número de delimitadores de abertura é maior do que o de fechamento. Por exemplo: ( [ ]
- c) Número de delimitadores de fechamento é maior do que o de abertura. Por exemplo: ( ( ) )

## Entrada

A primeira linha da entrada contém um valor inteiro N, que indica o número de expressões a ser avaliadas. Cada expressão a seguir possuirá, no máximo, 200 caracteres.

## Saída

Para cada expressão lida, apresente o resultado com base nos seguintes códigos:

- 1: Expressão correta
- -1: Erro de incompatibilidade
- -2: Erro de número de delimitadores de abertura maior que de fechamento
- -3: Erro de número de delimitadores de fechamento maior que de abertura

## Exemplos

### Entrada

```
9
{10 - [(5 + 6) * 10] + 20}
{10 - [(5 + 6) * 10} + 20]
((5+2) * (7*3)
(5+2)) * (7*3)
((( [{ [{ [{ } ] } ] } ] ) ) )
[ ] [ ] [ ] [ ] [ ] [ ]
(((( ( ) )))
[[ [ [ ] ] ] ]
(( [{ [{ [{ } ] } ] } ] ) )
```

## Saída

```
1
-1
-2
-3
1
1
-2
-3
-1
```