

Para efeitos da nota atribuída à resolução de exercícios ao longo do semestre - **Submeter até 23:59 de 20 de Novembro**
(o problema continuará depois disponível para submissão, mas sem contar para a nota)
[para perceber o contexto do problema deve [ler o guião da aula #07](#)]

[ED007] Entre Parênteses

Neste problema deverá submeter uma classe **ED007** contendo um programa completo para resolver o problema (ou seja, com o método `main`).

Pode assumir que no Mooshak terá acesso às classes de listas, pilhas e filas como dadas nas aulas (ou seja, não precisa de as incluir no código submetido).

O problema

A Joana estava a fazer um exercício de matemática quando se deparou com a seguinte expressão:

```
( 1 + ( 2 + 3 * [ 4 + 5 * ( 6 * ( 7 ) ) ] ) )
```

"Tantos parênteses!", pensou ela. E antes de começar a fazer exercício pensou também: "E se o professor pensou mal antes de começar a fazer o exercício e não colocou bem os parênteses. Se assim for nem vale a pena fazer o exerício". É tão preguiçosa esta Joana.



Como a Joana acredita em ti, tens de tentar fazer um programa que indique que a expressão matemática está mesmo bem formada, para que ela tenha vontade de fazer o exercício.

A tua tarefa é portanto fazer um analisador de expressões que ligue apenas aos parênteses. Tens de conseguir dizer se os parênteses estão bem formados, e caso contrário, indicar qual a posição onde foi encontrado o primeiro erro. E não é preciso explicar quando uma expressão com parênteses está bem formada, pois não?

Input

A primeira linha contém um número **N**, indicando o número de expressões a analisar.

As seguintes **N** linhas definem expressões matemáticas contendo:

- Dígitos, representando números inteiros
- Caracteres, representando operações: `+`, `-`, `*` ou `/`
- Parênteses curvos ou rectos: `(`, `)`, `[` ou `]`
- Espaços separando as várias partes da expressão

Não podes assumir nenhum tamanho máximo para a expressão, mas é garantido que existem espaços a separar todas as componentes da expressão.

Output

Uma linha para cada expressão, indicando **Expressao bem formada** caso todos os parenteses estejam bem emparelhados, **Erro na posicao POS**, caso seja descoberto um parênteses que não poderia aparecer naquele sítio (onde **POS** indica a posição na linha, começando a contar de zero, desse mesmo parênteses que está incorrecto) ou **Ficam parenteses por fechar** caso chegando ao fim da expressão fiquem ainda parenteses por fechar.

Note-se que por posição entende-se o número de caracteres até chegar ao sítio em causa. Por exemplo, na linha "`(1 + 2)`", o `(` está na posicao 0, o `+` na posição 4 e o `)` na posição 8.

Lembra-te que podes ignorar todo o conteúdo da expressão que não sejam parenteses, pois para este problema em particular, é irrelevante.

Exemplo de input/output

Input	Output
5 <code>(1)</code> <code>(1 + 2))</code>	Expressao bem formada Erro na posicao 10 Erro na posicao 4

[(])
(1 + (2)
(1 + (2 + 3 * (4 + 5 * (6 * (7)))))

Ficam parenteses por fechar
Expressao bem formada