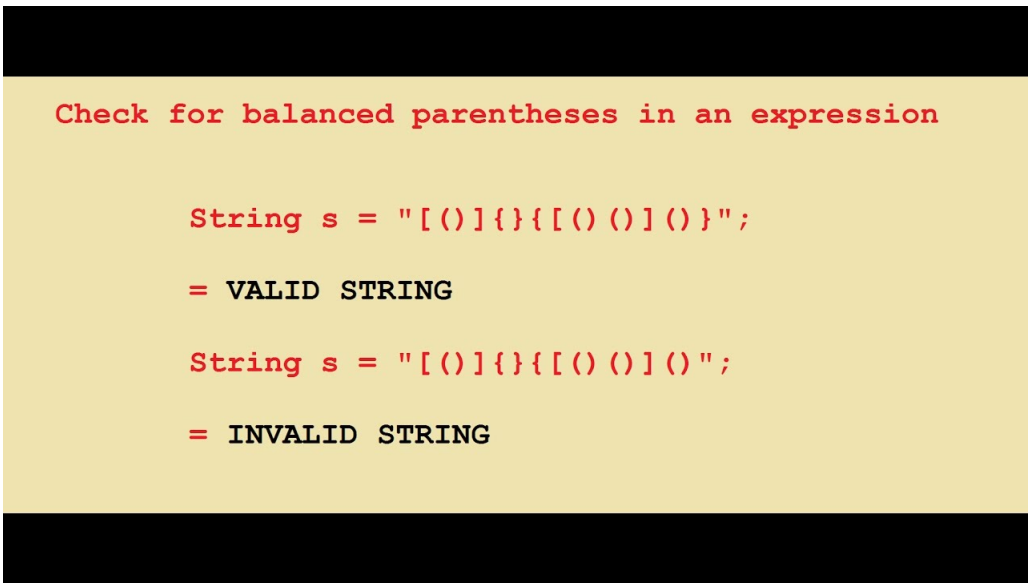

7 Expressões

 (++++)

Pedrinho e Zezinho precisam estudar a respeito da resolução de expressões matemáticas para uma prova que irão, em breve, prestar. Para isso, eles querem resolver muitos exercícios antes desta prova. Como sabem programar, então decidiram elaborar um "gerador de expressões matemáticas" – GEM – em $\mathbb{C}$.
O gerador de expressões que eles criaram funciona em duas fases:

1ª fase – É gerada uma cadeia de caracteres c_i que contém apenas os caracteres {, [, (, },] e). Ou seja: abre chave, abre colchete, abre parêntese, fecha chave, fecha colchete e, por fim, fecha parêntese.

2ª fase – O gerador adiciona números e operadores na estrutura criada na primeira fase, tendo por objetivo formar expressões aritméticas.

Ao terminar sua execução, o GEM produz uma cadeia de caracteres c, que pode ser dita "bem definida" (ou válida) se atender às seguintes propriedades:

1. $c = \lambda$, ou seja, c é uma cadeia de caracteres vazia (não contém nenhum caractere).
2. c é formada por uma cadeia "bem definida" envolvida por pares "abre" e "fecha" de parênteses, colchetes ou chaves.
   Portanto, se a cadeia S é "bem definida", então as cadeias (S), $[S]$ e S também são bem definidas.
3. c 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 GEM, pois algumas das cadeias geradas **não eram** do tipo "bem definida".
Eles querem começar a resolver as expressões o mais rápido possível, e sabendo que você é um ótimo programador, resolveram pedir sua ajuda: 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 cadeias e, por isso, a primeira linha da entrada contém um inteiro t indicando o número de cadeias – $1 \leq t \leq 20$. Em seguida tem-se t linhas, cada uma com uma cadeia c. Sabe-se que o tamanho de c pode variar de 1 (um) a 1000 (mil) caracteres.

Saída

Para cada instância da entrada, imprima uma linha contendo a palavra “bem-formada” se a cadeia é deste tipo ou “mal-formada” caso contrário.

Exemplos

Entrada	Saída
12 ( ) ([]) { } (] } { ([{ }]) { } [ ] ( ) ( )) { [ ] ((({ } { }()[[]])() { }) { } ((((((((((({ ([]) }]))))))))))	bem-formada bem-formada bem-formada mal-formada mal-formada bem-formada bem-formada mal-formada mal-formada mal-formada bem-formada mal-formada

Entrada	Saída
3 (()]]][[(())	mal-formada mal-formada bem-formada

Observação: Perceba todas as letras das palavras *bem-formada* e *mal-formada* estão grafadas em minúscula. Você deve fazer com que seu programa grafe *exatamente* desta maneira.