

<b>Curso: Licenciatura em Computação</b>	
<b>Disciplina:</b> Estruturas de Dados	<b>Professor:</b> Edson Angoti Júnior
<b>Instruções para entrega:</b> <b>1. Postar os códigos dos exercícios no Github</b> <b>2. Entregar a tarefa no Google Classroom indicando o link do repositório Github</b> <b>3. Exercícios retirados do livro texto do Tenenbaum</b>	

1. Simule a ação do algoritmo de análise de expressões para cada uma das seguintes strings, apresentando o conteúdo da pilha em cada ponto.

a. (A + B} )  
push ( '(' )      inválido porque não tem abertura '{ '  
i = pop ( '}' )

b. {[A+B]-[(C-D)]  
push ( '{ '  
push ( '[' '  
push ( '[' '  
push ( '(' '  
pop ( ) não tem fechamento } inválido

c. (A+B)-{C+D}-[F+G]  
push ( '(' '  
push ( '{ '  
push ( '{ '  
pop ( ')' )  
pop ( '}' )  
pop ( '[' ) válido tem todas aberturas e fechamentos

d. ((H)\*{([J+K]))}  
push ( '(' '  
push ( '(' '  
push ( '{ '  
push ( '(' '  
push ( '[' '  
pop ( ')' ) não tem o fechamento está inválido

e. (((A))))  
push ( '(' '  
push ( '(' '  
push ( '(' '  
pop ( ')' )  
pop ( ')' )  
pop ( ')' )  
pop ( ')' ) inválido não tem abertura

Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma:

```
x C y
ABABBA C ABBABA
push ('A')
push ('B')
push ('A')
push ('B')
push ('B')
push ('A')
pop ('A')
pop ('B')
pop ('B')
pop ('A')
pop ('B')
pop ('A')
```

A expressão vai ser válida, bem formada.

onde x é uma string consistindo nas letras 'A' e 'B', e y é o inverso de x (isto é, se x = "ABABBA" então y = "ABBABA"). Em cada ponto você poderá ler o próximo caractere da string.