Aula Prática 6 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho
Universidade Federal de Ouro Preto
Departamento de Computação

7 de julho de 2021

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

1 Manipulação de pilhas

Dada uma expressão qualquer com parênteses, indique se a quantidade de parênteses está correta ou não, sem levar em conta o restante da expressão. Por exemplo:

- `a+(b*c)-2-a` está correto
- `(a+b*(2-c)-2+a)*2` está correto

enquanto

- `(a*b-(2+c)` está incorreto
- `2*(3-a))` está incorreto
- `)3+b*(2-c)(` está incorreto

Ou seja, todo parênteses que fecha deve ter um outro parênteses que abre correspondente e não pode haver parênteses que fecha sem um prévio parênteses que abre e a quantidade total de parênteses que abre e fecha deve ser igual.

Especificação da Entrada

Como entrada, haverá uma expressã de tamanho arbitrário, sem que haja espaços em branco.

Especificação da Saída

Como saída, para cada expressão de entrada deverá ser gerado uma linha indicando o resultado do processamento, cada uma delas contendo as palavras **correto** ou **incorreto** de acordo com as regras acima fornecidas.

Exemplo de Entradas

```
a+(b*c)-2-a
(a+b*(2-c)-2+a)*2
(a*b-(2+c)
2*(3-a))
)3+b*(2-c)(
```

Exemplo de Saídas

```
correto
correto
incorreto
incorreto
incorreto
```

Estrutura do código

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Uma pilha de caracteres deve ser manipulada para determinação do resultado. Note que deve ser respeitada a política de operações em pilhas, não devendo ser implementadas operações inválidas, como percorrer a pilha. As funções *Pilha_Inicia*, *Pilha_Push*, *Pilha_Pop*, *Pilha_EhVazia* são as mesmas vistas na aula teórica com modificações mínimas para tratar caracteres. A função *Pilha_Esvazia* é a única que precisa ser criada para resolução deste problema.

Diretivas de Compilação

```
$ gcc pilha.c -c
$ gcc principal.c -c
$ gcc pilha.o principal.o -o programa
```