

AULA PRÁTICA 04

- **Data de entrega: Até 02 de maio às 23:59:59.**

- **Procedimento para a entrega:**

1. Submissão: via **run.codes**.
2. Os nomes dos arquivos e das funções devem ser especificados considerando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos `.h` e `.c` sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém `main()`, devem ser compactados (`.zip`), sendo o arquivo resultante submetido via **run.codes**.
6. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.
7. Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado.
8. Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas.
9. A avaliação considerará o tempo de execução e o percentual de respostas corretas.
10. Eventualmente, serão realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação.
11. Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada.
12. Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software.
13. 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.
14. Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos.
15. Não serão considerados algoritmos parcialmente implementados.

- **Bom trabalho!**

Atividade

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

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 e da saída

Como entrada, haverá uma expressão de tamanho arbitrário, sem que haja espaços em branco. Como saída, para a expressão de entrada deverá ser gerado uma linha indicando o resultado do processamento: correto ou incorreto.

Entrada	Saída
$a + (b * c) - 2 - a$	correto

Entrada	Saída
$(a * b - (2 + c))$	incorreto

Diretivas de Compilação

```
$ gcc -c lista.c -Wall
$ gcc -c pilha.c -Wall
$ gcc -c pratica4.c -Wall
$ gcc lista.o pilha.o pratica4.o -o exe
```

Considerações

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Você primeiro irá implementar as funções de uma lista (TListaInserePrimeiro, TListaRetiraPrimeiro, TListaImprime, TListaInicia, TListaEhVazia e TListaEsvazia). Depois, você irá implementar e manipular uma pilha de caracteres para verificar se a quantidade e a ordem dos parênteses estão corretas. Note que **uma pilha nada mais é que uma lista com um protocolo LIFO bem definido**. A política de operações em pilhas deve ser respeitada, não devendo ser implementadas operações inválidas, como percorrer a pilha. As funções TPilhaInicia, TPilhaPush, TPilhaPop, TPilhaEhVazia são as mesmas vistas na aula teórica com modificações mínimas para tratar caracteres. A função TPilhaEsvazia é a única que precisa ser criada para resolução deste problema. Observe também que a implementação das funções da pilha devem ser feitas usando as funções de lista.

- Não altere o nome dos arquivos.
- O arquivo .zip deve conter na sua raiz somente os arquivo-fonte (lista.h, lista.c, pilha.h, pilha.c e pratica4.c).
- Há no total **cinco** casos de teste. Você terá acesso (entrada e saída) a **dois** deles para realizar os seus testes.
- Onde há ????? você deve substituir por um tipo primitivo ou outro TAD.
- A implementação das funções da pilha devem ser feitos usando as funções da lista.