

1 Sobre a entrega do trabalho

São requisitos para atribuição de notas a este trabalho:

- Uso de um arquivo makefile para facilitar a compilação. Os professores rodarão “make” e deverão obter o arquivo executável funcional com a sua solução. Este executável, cujo nome deverá ser tp3, deverá estar no subdiretório tp3;
- Ao compilar, incluir pelo menos `-Wall -Werror -Wextra`. Se não compilar, o trabalho vale zero. Haverá desconto por cada *warning*;
- Arquivo de entrega:
 - Deve estar no formato tar comprimido (.tar.gz);
 - O tar.gz deve ser criado considerando-se que existe um diretório com o nome do trabalho. Por exemplo, este trabalho é o tp3;
 - Então seu tar.gz deve ser criado assim:
 - * Estando no diretório tp3, faça:
 - * `cd ..`
 - * `tar zcvf tp3.tar.gz tp3`
 - Desta maneira, quando os professores abrirem o tar.gz (com o comando `tar zxvf tp3.tar.gz`) terão garantidamente o diretório correto da entrega para poderem fazer a correção semi-automática.
 - O que colocar no tar.gz? Todos os arquivos que são necessários para a compilação, por isso se você usa arquivos além dos especificados, coloque-os também. Mas minimamente deve conter todos os arquivos `.c`, `.h` e o makefile;
 - Os professores testarão seus programas em uma máquina do departamento de informática (por exemplo, cpu1), por isso, antes de entregar seu trabalho faça um teste em máquinas do dinf para garantir que tudo funcione bem.

2 O trabalho

Você deve baixar o `tp3.tar.gz` anexo a este enunciado e abri-lo para poder fazer o trabalho, pois irá precisar de todos os arquivos ali contidos:

libpilha.h: arquivo (read only) de *header* com todos os protótipos das funções para manipular a pilha;

testa_pilha.c: arquivo (read only) para testar sua implementação do `libpilha.h`.

makefile: sugestão de um makefile que você pode usar (ou adaptar, se quiser).

Os arquivos `libpilha.h` e `testa_pilha.c` não podem ser alterados. Na correção, os professores usarão os arquivos originais.

- Use boas práticas de programação, como indentação, bons nomes para variáveis, comentários no código, bibliotecas, defines... Um trabalho que não tenha sido implementado com boas práticas vale zero.
- Quaisquer dúvidas com relação a este enunciado devem ser solucionadas via email para `prog1prof@inf.ufpr.br` pois assim todos os professores receberão os questionamentos. Na dúvida, não tome decisões sobre a especificação, pergunte!
- Dúvidas podem e devem ser resolvidas durante as aulas.

3 O problema

Este trabalho consiste na implementação do programa `libpilha.c` com base no `libpilha.h` fornecido.

Sua implementação deve usar uma lista ligada com nodo cabeça que implementa a política LIFO (Last In First Out), isto é, uma política que implementa um Tipo Abstrato de Dados **Pilha**.

O desafio é aprender a usar alocação dinâmica e manter a lista sempre ligada, sem perder ponteiros.

Para testar sua implementação, use o arquivo `testa_pilha.c` fornecido. Procure entender a motivação de cada teste feito e a sua importância.

Isso é muito importante para o restante do curso, por isso aprenda o conceito e os detalhes semânticos e sintáticos o quanto antes!

Bom trabalho!