

Estruturas de Dados

Trabalho Prático I - Pilhas

1. Regras

- O trabalho deverá ser desenvolvido individualmente.
- O trabalho deve ser entregue através do Moodle até as 23:59 horas do dia 09/09. Os códigos dos programas desenvolvidos devem ser postados, em um arquivo compactado (.zip) e o nome do arquivo deve ser `trab1_nomeAluno`. Por exemplo, caso o trabalho tenha sido feito por João, deve ser enviado um arquivo com o nome `trab1_joao.zip`.
- Devem estar compactados no .zip os dois programas principais (.c) e as duas bibliotecas de pilhas (.c e .h).
- O trabalho vale 10 pontos.
- Trabalho copiado ou entregue fora do prazo vale 0(zero).

O professor poderá realizar arguição individual sobre o trabalho a fim de averiguar a compreensão de cada aluno sobre o trabalho realizado podendo a nota do trabalho ser substituída pela nota na arguição. A arguição será especialmente importante em casos de suspeita de cópias.

2. Introdução:

Os objetivos principais deste trabalho são:

- Implementar duas bibliotecas da estrutura de dados pilha, uma utilizando vetor e outra utilizando ponteiros (alocação dinâmica de memória)
- Praticar a reutilização, utilizando as mesmas bibliotecas para implementar dois programas: um que faça a conversão de números decimais para binários e um que implemente o jogo “Torre de Hanói”.
- Praticar o encapsulamento: as duas bibliotecas devem ter as mesmas interfaces (métodos com mesma assinatura) de forma que os programas principais não precisem ser alterados quando a biblioteca importada é alterada.

3. Requisitos Funcionais do Programa de transformação de decimal para binário

A transformação de números da base decimal para a binária envolve seguidas divisões inteiras por 2 armazenando os restos dessas divisões, conforme exemplificado em sala e já estudado em outras disciplinas.

O programa a ser desenvolvido deve solicitar que o usuário digite um número decimal inteiro positivo para transformar em binário ou um número não positivo para finalizar a execução. Caso o usuário digite um número positivo, o programa deve convertê-lo para

base binária e solicitar a digitação de um novo número, ficando nesse laço até que seja digitado um número não positivo.

O programa deve utilizar pilhas para fazer a transformação.

4. Requisitos Funcionais do Programa “Torre de Hanói”

As **Torres de Hanói** são um [quebra-cabeças](#) que consiste em uma base contendo três pinos, onde num deles, são dispostos n [discos](#) uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo. O problema consiste em passar todos os discos de um pino para outro qualquer, usando um dos pinos como auxiliar, de maneira que um disco maior nunca fique em cima de outro menor em nenhuma situação. O número de discos pode variar sendo que o mais simples contém apenas três. As Torres de Hanói tem sido tradicionalmente considerada como um procedimento para avaliação da capacidade de memória de trabalho, e principalmente de planejamento e solução de problemas. (http://pt.wikipedia.org/wiki/Torre_de_Hanoi).

Você deve desenvolver em C um programa que implemente o jogo Torre de Hanoi.

Cada pino do jogo será representada por uma pilha. Os discos serão elementos das pilhas. Como cada disco deve ter um tamanho, representado através de números inteiros indo de 1 a 5 onde 5 é o número de discos a serem utilizados no jogo. Assim, as informações a serem armazenadas em nossas pilhas são números inteiros.

O programa deve exibir o estado dos três pinos e solicitar que o usuário informe o pino de origem e o pino de destino do seu próximo movimento. Então, o programa deve avaliar se o movimento solicitado é permitido, seguindo a regra da torre de Hanói de nunca colocar um disco maior sobre um menor. Se o movimento solicitado for válido, o sistema deve efetuar tal movimento e atualizar o estado dos pinos. Caso seja inválido, o sistema deve informar ao usuário que o movimento é inválido e não realizar alterações nos estados dos pinos.

O programa deve ficar neste laço até que o usuário consiga chegar ao objetivo de mover todos os discos para o último pino. Quando isso acontecer o programa deve informar quantos movimentos o usuário fez até chegar ao objetivo final.

Requisito Bônus: Esse requisito não é obrigatório, mas será valorizado: implementar o jogo de forma que o usuário possa informar o número de discos que deseja. Nesse caso, o primeiro passo do usuário será informar a quantidade de discos desejados sendo, no

mínimo 3 discos. Quando o usuário informar o número de discos (n), os discos devem ser inseridos no primeiro pino (pilha).

5. Requisitos não funcionais:

Devem ser implementadas duas bibliotecas de pilha (uma com vetores e outra com ponteiros), contendo ao menos os seguintes métodos:

- Iniciar pilha;
- Empilhar;
- Desempilhar;
- Verificar se pilha está vazia;
- Verificar se pilha está cheia;
- Consultar elemento do topo da pilha;
- Imprimir todos elementos da pilha;

Os métodos das duas bibliotecas devem ter assinaturas iguais de forma que as bibliotecas possam ser substituídas sem afetar o código fonte.

Ademais, os dois programas desenvolvidos devem usar a biblioteca de pilha.

5. Critérios de avaliação

Além da adequação aos requisitos acima listados, serão considerados na avaliação:

- Modularidade do código (divisão do código em funções e procedimentos);
- Legibilidade do código: indentação, padrões de nomes de variáveis;
- Qualidade dos comentários.