

Universidade Tecnológica Federal do Paraná – Campus Apucarana
Estruturas de Dados (ED62A) – 2º semestre
Prof. Dr. Rafael Gomes Mantovani

Instruções:

- A interpretação dos enunciados faz parte da avaliação;
- É permitido o uso de rascunho no auxílio das questões. Contudo, os rascunhos devem ser identificados e entregues junto com a prova;
- É permitido consultar a folha com os tipos definidos e as assinaturas dos métodos que as manipulam.
- As respostas das questões que envolvem o desenvolvimento de código devem ser elaboradas em C. Contudo, soluções em pseudocódigo podem ser avaliadas em até metade da nota do exercício;
- As respostas/resoluções das questões devem ser escritas com caneta (preta/azul).

Exercício 1. (2,00 pontos) Na resolução de um problema por meio de um programa, a primeira providência é conceber um algoritmo adequado. A eficiência de um algoritmo está intimamente relacionada à disposição, na memória, dos dados que são tratados por um programa. Desta forma, faça um comparativo entre os diferentes tipos de estruturas de dados elementares. Lembre-se também de analisá-las levando em conta o tipo de implementação.

Exercício 2. (2,0 pontos) Um depósito guarda caixas de madeira em pilhas numeradas que são distribuídas por um grande galpão. Para gerenciar as informações sobre o armazenamento, o sistema de administração do depósito utiliza o tipo abstrato de dados **Pilha**, que implementa as funções descritas na tabela a seguir:

Função	Descrição
<code>void iniciaPilha(Pilha *pilha);</code>	Inicia uma nova pilha alocada dinamicamente
<code>int tamanho(Pilha *pilha);</code>	Consulta o número de elementos de uma pilha.
<code>Caixa* topo(Pilha *pilha);</code>	Consulta o elemento no topo de uma pilha.
<code>Caixa* pop(Pilha *pilha);</code>	Retira um elemento do topo de uma pilha.
<code>void push(Pilha *pilha, Caixa *caixa);</code>	Insere um elemento no topo de uma pilha.
<code>void destroi(Pilha *pilha);</code>	Esvazia e libera a memória alocada para uma pilha.

Tabela 1: Operações básicas com uma pilha dinâmica

O tipo de dados **Pilha** foi declarado para armazenar ponteiros para variáveis **Caixa**, cujo: campo **peso** armazena o peso da caixa, o campo **nome** contém o nome do objeto guardado na caixa, e o índice guarda o código único daquele produto. O sistema mantém um vetor para o tipo de dados **Pilha**, em que cada elemento armazena uma pilha de caixas no galpão. Sendo assim, responda às seguintes questões:

- a) **(0,5 pontos)** Defina o(s) tipo(s) abstrato(s) de dados em C necessário(s) para a definição de uma pilha que atue no sistema e armazene as Caixas do depósito;
- b) **(1,5 pontos)** Escreva uma função em C para empilhar uma nova caixa escolhendo a primeira pilha - na ordem do vetor -, de tal forma que a caixa não seja colocada sobre uma mais leve e que cada pilha tenha no máximo 10 caixas. Os parâmetros da função são o vetor de pilhas, o tamanho do vetor, e o ponteiro ***caixa** para a nova caixa a ser empilhada. O retorno da função deve ser o índice que indica a pilha onde a caixa foi empilhada, ou -1 se não foi possível empilhar a caixa obedecendo aos critérios desejados. A função tem o seguinte protótipo:

```
int empilhaCaixa(Pilha *deposito, int n, Caixa* caixa);
```

Exercício 3. (2,00 Pontos) Considere uma área de armazenamento de dados em um computador com **3 espaços** físicos. Sobre esta área está montada uma fila de nome **FILA**. Qual o comportamento da fila durante as seguintes operações descritas na tabela 2 ? Desenhe e identifique o estado da fila **em cada** uma das operações.

Tabela 2: Sequência de operações em um tipo Fila.

Id	Operação
1	Inserção de um nó com valor VERMELHO
2	Inserção de um nó com valor VERDE
3	Inserção de um nó com valor AMARELO
4	Inserção de um nó com valor BRANCO
5	Inserção de um nó com valor PRETO
6	Remoção de um nó
7	Remoção de um nó
8	Inserção de um nó com valor ROSA
9	Remoção de um nó
10	Remoção de um nó
11	Inserção de um nó com valor AZUL
12	Inserção de um nó com valor CINZA
13	Remoção de um nó
14	Remoção de um nó

Exercício 4. (2,00 Pontos) Diferente de filas e pilhas, uma *deque* (Doble-ended queue, ou Fila de extremidade dupla em português) permite inserção e remoção em ambas as extremidades.

- (0,5 pontos)** Defina o(s) tipo(s) abstrato(s) de dados para que se possa implementar e manipular uma *deque* **dinâmica**.
- (1,5 pontos)** Descreva procedimentos (em C) para inserir elementos em ambas as extremidades de uma **deque** (fila de duas extremidades).

Exercício 5. (2,00 Pontos) Implemente uma função que receba um vetor de valores inteiros com n elementos e construa uma lista encadeada armazenando os elementos do vetor nos nós da lista. Se o vetor tiver zero elementos, a função deve ter como valor de retorno uma lista vazia. O protótipo da função é dado por:

```
Lista* constroi(int n, int* v);
```