

Lista de Exercícios

Pilhas:

1) Empilhamento decrescente. Uma empilhadeira carrega caixas de 7, 5, 3 toneladas. Há três pilhas A, B, C. A pilha A é onde se encontram todas as caixas que chegam no depósito. Com um detalhe: caixas maiores não podem ser empilhadas sobre caixas menores. Elabore uma função chamada `chegaNoDeposito` (`Caixa *pNova`, `tPilha *pPilha`) que efetue o controle das caixas, de forma que caso uma caixa de maior peso do que uma que já está em A deva ser empilhada, então, todas as caixas que estão em A são movidas para as pilhas auxiliares B (contendo somente caixa de 5 toneladas) e C (contendo somente caixas de 3 toneladas) até que se possa empilhar a nova caixa. Depois, todas as caixas são movidas de volta para a pilha A. Crie e utilize as funções com os seguintes protótipos:

```
int vazia(tPilha *pPilha);
```

```
int cheia(tPilha *pPilha);
```

```
void empilhar(Caixa *pNova, tPilha *pPilha);
```

```
void desempilhar(tPilha *pPilha);
```

```
Caixa *topo(tPilha *pPilha);
```

2) Utilizando uma pilha, escreva um método que receba um número inteiro positivo no formato decimal e converte este número para o formato binário.

Exemplos:

5 = 101

13 = 1101

1 = 1

O método retorna uma lista encadeada com os dígitos que fazem parte do número em binário. Considere $x \text{ DIV } y$ e $x \text{ MOD } y$ o quociente e o resto da divisão entre dois números inteiros x e y , respectivamente, na linguagem algorítmica.

3) Faça um programa que leia uma seqüência de caracteres terminada por <Enter> e a exiba criptograficamente segundo as seguintes regras:

- Toda não consoante é exibida diretamente
- Toda seqüência de consoantes é exibida na ordem inversa obtida

Lista:

1) Escreva o programa para fazer o controle dos sapatos de uma sapataria. Para cada sapato são guardadas as seguintes informações: código, nome (do modelo), quantidades por tamanho (4 quantidades, uma para cada tamanho do 35 até o 38) e ano.

Inicialmente, construa uma lista de sapatos ordenada por código, sendo que essa lista deve ser mantida ordenada já durante a sua construção. Em seguida, deve ser apresentado um menu de opções, sendo lidas e tratadas as seguintes opções:

V: verifica se a lista esta ordenada por ano;

S: separa os sapatos da lista em duas novas listas, uma com os sapatos de ano <2000 e outra com os sapatos de ano >= 2000, mostrando-os na tela;

M: percorre a lista, mostrando os sapatos na tela;

I: lê os dados de um novo sapato, inserindo-o na lista;

R: lê um código de sapato, retira o sapato correspondente da lista, e apresenta as informações do sapato retirado na tela;

F: fim.

2) Dada uma lista ordenada, escreva as seguintes funções

- a. verifique se a lista está ordenada e retorna 0 se estiver desordenada, -1 se em ordem decrescente ou 1 se em ordem crescente.
- b. copia a lista em outra lista também recebida como parâmetro, inicialmente vazia
- c. copia a lista em outra lista também recebida como parâmetro, inicialmente vazia, eliminando repetidos
- d. inverte a lista armazenando o resultado em outra lista inicialmente vazia recebida como parâmetro
- e. inverte a lista armazenando o resultado na própria lista
- f. gera uma lista também recebida como parâmetro, inicialmente vazia, onde cada registro contém, além dos dados da primeira lista, um campo inteiro representando a quantidade de vezes que o elemento apareceu na lista
- g. elimina da lista todas as ocorrências de um dado elemento. Considere a lista desordenada
- h. elimina da lista todas as ocorrências de um dado elemento. Considere a lista ordenada

Filas

- 1) Dada uma fila de inteiros, escreva um programa que exclua todos os números negativos sem alterar a posição dos outros elementos da fila.
- 2) Faça uma função que receba três filas, duas já preenchidas em ordem crescente e preencha a última com os valores das duas primeiras em ordem crescente.
- 3) Como você implementaria uma fila de pilhas? Uma pilha de filas? Uma fila de filas? Escreva rotinas para implementar as operações corretas para cada uma destas estruturas de dados.

Exercícios extraídos de (Referências)

[1] Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein, Estruturas de Dados Usando C, Makron Books/Pearson Education, 1995.

[2] N. Ziviani, F.C. Botelho, Projeto de Algoritmos com implementações em Java e C++, Editora Thomson, 2006.