

Universidade de Brasília – UnB
Faculdade de Tecnologia – FT
Departamento de Engenharia Elétrica – ENE
Disciplina: Algoritmos e Estrutura de Dados
Professor: Vinícius Pereira Gonçalves
Aluno(a):



Matrícula:

2ª Avaliação

REGRAS:

- 1 – A prova deverá ser resolvida em C++;
- 2 – Incluir comentários no seu código, com o objetivo mostrar onde você resolveu cada item solicitado nas questões;
- 3 – No caso de o aluno tentar copiar informações ou respostas da prova de um outro aluno, ambos serão desclassificados da avaliação, sendo-lhes atribuída a menção SR.
Regra aplicada tanto para meios analógicos quanto digitais. Programas copiados da internet também terão menção SR.
- 4 – Deverá ser mantido silêncio total e absoluto durante a realização da prova.
- 5 – A interpretação das questões é parte integrante da avaliação.
- 6 – Crie um arquivo .cpp para cada questão e ao final da prova faça upload de um arquivo .zip das questões no Campus Virtual.

QUESTAO 01 (3 pts) – Um banco privado separa seus clientes por categorias embasadas na movimentação mensal de dinheiro em suas contas, a saber, temos os clientes Silver (1ª categoria, com movimentação de até R\$ 3.000,00), clientes Gold (2ª categoria, com movimentação entre R\$ 3.000,00 e 100.000,00) e os clientes Black (3ª categoria, com movimentação maior que R\$ 100.000,00).

Todos os dias o banco atende cerca de 20 clientes que vão até a agência. Chegando ao banco os clientes devem informar o Nome, Idade e selecionar se sua conta é Silver, Gold ou Black.

A lei brasileira define que todas as pessoas acima de 60 anos de idade devem ter uma fila preferencial. O banco também tenta sempre dar preferência a todos os clientes Black, depois os Gold e por último os clientes Silver. A ordem de atendimento deve ser sempre:

1º – Clientes acima de 60 anos (entre eles, devem estar na seguinte ordem - 1º Black, 2º Gold, 3º Silver).

2º – Clientes Black abaixo de 60 anos.

3º – Clientes Gold abaixo de 60 anos.

2º – Clientes Silver abaixo de 60 anos.

Construa um Programa para ORDENAR todos esses 20 clientes em uma FILA ORDENADA de acordo com a descrição supramencionada, com preenchimento pelo usuário.

Além disso, você deve ter uma LISTA secundária ORDENADA por ordem alfabética, pois o departamento de recursos humanos do banco sempre usa os dados dos clientes atendidos para análises estatísticas.

Seu programa deve conter um MENU:

- 1- Cadastrar cliente (total de 20)
- 2- Mostrar ordem da fila de atendimento.
- 3- Mostrar todos os clientes em ordem alfabetica.
- 4- Sair.

OBS: Para realizar a Ordenação usar: **Quick sort ou Bubble sort ou Merge sort ou Selection sort.**

Exemplo para 5 clientes:

MENU

ENTRADA DE CADASTRO:

João

30 anos

Black

Marcos

60 anos

Black

Maria

65 anos

Silver

Joana

45 anos

Silver

Marcelo

18 anos

Gold

SAIDA, FILA DE ATENDIMENTO:

1- Marcos . Black

2- Maria . Silver

3- João . Black

4- Marcelo . Gold

5- Joana . Silver

SAIDA, ORDEM ALFABETICA:

1- Joana . 45 anos . Silver

2- João . 30 anos . Black

3- Marcelo . 18 anos . Gold

4- Maria . 65 anos . Silver

5- Marcos . 60 anos . Black

QUESTAO 02 (3 pts) – Construa um programa que implemente uma TABELA HASH de tamanho $M=11$, com endereçamento aberto utilizando a função $h(k) = k \bmod M$. (mod é uma função para resto da divisão ex: $13 \bmod 11 = 2$).

- Mostre a configuração da tabela após a inserção dos registros com as chaves: 4, 17, 13, 35, 25, 11, 2, 10, 32, 55, 64, 23, 3, 22.
- Mostre a configuração da tabela após a remoção dos registros com as chaves: 25, 11.
- Mostre a configuração da tabela após a inserção dos registros com as chaves: 40, 3.

Você deve trabalhar com LISTAS DUPLAMENTE ENCADEADAS para construir a tabela.

O programa deve conter um menu com os seguintes itens:

- 1- Inserção de elementos.
- 2- Remoção de elementos.
- 3- Mostrar a configuração da tabela.
- 4- Sair

```
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.

1
digite o numero de chaves que deseja inserir
15
4 17 13 35 25 11 2 10 32 55 64 23 3 22
44
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.

3
0 [ 11 ] [ 55 ] [ 22 ] [ 44 ] --> \
1 [ 23 ] --> \
2 [ 13 ] [ 35 ] [ 2 ] --> \
3 [ 25 ] [ 3 ] --> \
4 [ 4 ] --> \
5 --> \
6 [ 17 ] --> \
7 --> \
8 --> \
9 [ 64 ] --> \
10 [ 10 ] [ 32 ] --> \
```

```
10 [ 10 ] [ 32 ] --> \
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.
```

2

quantas chaves deseja retirar?

3

digite as chaves que deseja retirar.

25 11 2

A CHAVE 25 FOI RETIRADA COM SUCESSO

A CHAVE 11 FOI RETIRADA COM SUCESSO

A CHAVE 2 FOI RETIRADA COM SUCESSO

```
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.
```

3

A CHAVE 2 FOI RETIRADA COM SUCESSO

```
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.
```

3

```
0 [ 55 ] [ 22 ] [ 44 ] --> \
```

```
1 [ 23 ] --> \
```

```
2 [ 13 ] [ 35 ] --> \
```

```
3 [ 3 ] --> \
```

```
4 [ 4 ] --> \
```

```
5 --> \
```

```
6 [ 17 ] --> \
```

```
7 --> \
```

```
8 --> \
```

```
9 [ 64 ] --> \
```

```
10 [ 10 ] [ 32 ] --> \
```

```
1) insercao dos elementos.
2) remocao dos elementos.
3) mostrar a configuracao da tabela.
4) sair.
```

QUESTAO 03 (3 pts) – Suponha uma LISTA DUPLAMENTE ENCADEADA com Nome e Idade de pessoas. Implemente um programa que adiciona dinamicamente cada pessoa e sua idade. O programa deve permitir uma BUSCA BINÁRIA pelo nome e retornar: a pessoa procurada, a pessoa anterior e a pessoa posterior à procurada. O programa deve também poder remover uma pessoa, entre outras funcionalidades dispostas a seguir:

- 1- Adicionar pessoa;
- 2- Remover pessoa;
- 3- Pesquisar pelo nome (mostrar seus vizinhos);
- 4- Mostrar todos em ordem alfabética;
- 5- Mostrar todos em ordem de idade;
- 6- Sair.

EX: Ana - 20, João - 30, Guilherme - 22, Joana - 21, Mariana - 28.

Pesquisar por João: Resposta -> anterior: Ana - 20. Posterior: Guilherme - 22.
Procurado: João - 30.

OBS: O programa deve usar o paradigma de **Programação Orientado a Objetos** e possuir pelo menos uma **classe**. Cada uma das funções descritas no menu deve estar em um **método** diferente na classe criada. Para realizar a Ordenação usar: **Quick sort ou Bubble sort ou Merge sort ou Selection sort**.