



Pontifícia Universidade Católica de Minas Gerais

ICEI – Instituto de Ciências Exatas e Informática

DCC – Departamento de Ciência da Computação

Campus Belo Horizonte – Unidade Praça da Liberdade

Bacharelado em Ciência da Computação

MAIOR UNIVERSIDADE CATÓLICA DO MUNDO - Fonte: Vaticano

MELHOR UNIVERSIDADE PRIVADA DO BRASIL - Guia do Estudante, por 6x

ENTRE AS MELHORES UNIVERSIDADES DO MUNDO – Times (Ranking Times High Education)

ÁREA DA COMPUTAÇÃO PUC MINAS: SEMPRE ENTRE AS 4 DO PAÍS NA

PREFERÊNCIA DO MERCADO (RH) AO CONTRATAR – Folha de São Paulo, RUF, desde 2012

CIÊNCIA DA COMPUTAÇÃO PUC MINAS: SEMPRE 4 OU 5 ESTRELAS - Guia do Estudante

Algoritmos e Estruturas de Dados I

Professor: Lúcio Mauro Pereira

Reavaliação – Valor: 100 pontos

18 de dezembro de 2023

A nota final corresponderá à média da nota obtida no semestre com aquela obtida nesta prova

Aluno(a): _____ Nota: _____

Antes de iniciar a avaliação, leia atentamente as instruções a seguir.

Desligue o seu celular e guarde-o.

A prova não poderá ser desgrampeada.

Prova **individual** e **sem consulta**.

Deixe sobre a carteira **apenas** caneta, lápis e borracha.

Não é permitido material algum sobre o **colo**, como blusas ou mochilas - guarde-os **debaixo da carteira**.

Não é autorizado o empréstimo de materiais durante a prova.

Caso alguma das regras acima seja violada, a prova será anulada.

Antes de desenvolver cada questão analise o problema, planeje uma solução, elabore um modelo de solução através de fluxogramas ou algoritmos. Em seguida, **codifique-o em C/C++**.

A correção incidirá apenas sobre a solução codificada.

A correção de cada questão considerará:

- o atendimento ao problema proposto;
- a qualidade da solução lógica;
- a codificação do programa;
- a indentação do código;
- a escolha adequada da estrutura de repetição;
- a documentação do programa.

Não haverá atendimento individual durante a prova. A interpretação das questões faz parte da avaliação. Fique à vontade para justificar, junto à questão, as decisões que tomar durante a solução.

Fique à vontade para usar o verso da folha sempre que necessário.

Tempo estimado para realização da prova: **70 minutos**.

Considere declarados os identificadores **MAX_LIN** e **MAX_COL**, representando, respectivamente, o número máximo de linhas e o número máximo de colunas.

Para arranjos **unidimensionais**, considere o tamanho máximo declarado em **MAX**.

Para **matrizes quadradas**, considere a dimensão definida por **MAX_LIN x MAX_LIN**.

1. $(20,0)$ Objeto de avaliação: **Estrutura de repetição, funções, arranjos**

Construa uma função com dois argumentos, uma matriz quadrada de reais e um valor inteiro x . O método deverá retornar *verdadeiro* se a linha x da matriz for igual à coluna x , e *falso* caso contrário.

2. $(20,0)$ Objeto de avaliação: **Funções recursivas, arranjos**

Crie uma versão recursiva para a questão anterior.

3. (20,0) Objeto de avaliação: **Est. de repetição, funções, arranjos**

Construa uma função que receba dois arranjos unidimensionais: um de caracteres e o outro de inteiros – este, de tamanho igual a cinco. O método deverá percorrer o arranjo de caracteres para contar o número de vogais nele presentes. A quantidade de cada uma das cinco vogais deverá estar representada no arranjo de inteiros, de forma que o primeiro elemento conte o número de ocorrências da vogal ‘a’, o segundo elemento o número de ocorrências da vogal ‘e’ e assim sucessivamente. Lembre-se que não é possível assumir o conteúdo prévio dos arranjos enviados ao método.

4. (40,0) Objeto de avaliação: **Classes, encapsulamento, generalização, membros estáticos**

A
- x : real (<i>pode haver outros</i>)
+ setA(real, real) + setX(real) + getX() : real (<i>pode haver outros</i>)

Implemente a(s) classe(s) envolvida(s) nas linhas de código abaixo (*pode haver outras classes além da classe A modelada ao lado*).

A(s) classe(s) deverão prover apenas (e todos) os atributos e métodos necessários para que as instruções abaixo possam ser executadas corretamente.

Analise atentamente todas as instruções antes de iniciar a codificação.

```
A *objeto = new A();
objeto->setA(3.5, 2.7);
objeto->escreve(); // escreve todos os atributos do objeto na tela do monitor de vídeo
```

```
cout << endl << A::quantidade(); // escreve o número de instâncias da classe A que já foram criadas
```

```
A *outro = new A(1.8, 2.5);
float m = objeto->maior(outro); // recupera o maior valor de X: o do objeto corrente ou daquele parametrizado
```

```
cout << endl << objeto->getY(); // escreve o valor de Y, sendo este herdado da classe B
```