



Pontifícia Universidade Católica de Minas Gerais

ICEI – Instituto de Ciências Exatas e Informática
DCC – Departamento de Ciência da Computação
lo Horizonte – Unidade Praça da Liberdade
Bacharelado em Ciência da Computação

Algoritmos e Estruturas de Dados I

Professor: Lúcio Mauro Pereira

Prova III – Valor: 35 pontos

13 de dezembro de 2023

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

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. Principais objetos de avaliação: **manipulação de arranjos e abordagens iterativa e recursiva.**

Considere os vetores de reais, A e B, ambos de tamanho físico igual a TAM. Uma função deverá calcular e retornar o número de elementos iguais em uma mesma posição nos dois vetores. Argumentos da função: os dois vetores objetos da comparação e o tamanho lógico de ambos os vetores. Ajuste os parâmetros quando julgar necessário.

- a) $(5,0)$ Abordagem iterativa
- b) $(5,0)$ Abordagem recursiva

2. Principais objetos de avaliação: **manipulação de arranjos (matrizes)**

(5,0) Na Teoria dos Sistemas, define-se o elemento MINMAX de uma matriz como sendo o maior elemento da linha onde se encontra o menor elemento da matriz. Construa uma função que receba uma matriz de reais e calcule e retorne seu MINMAX. A tarefa desta função deverá ser distribuída em outras funções.

3. Principal objeto de avaliação: **Princípios da Orientação por Objetos**

Um veículo pode se especializar em carro ou em moto.

São propriedades comuns a carro e moto: modelo, uma *string*; ano de fabricação, um valor inteiro.

Quando um veículo se especializa em carro, acrescenta-se o atributo *tração*, uma *string*.

Quando um veículo se especializa em moto, acrescenta-se o atributo *cilindradas*, um valor inteiro.

Considere as seguintes instruções:

```
Veiculo* veiculo[4];  
veiculo[0] = new Carro("Gol", 2020, "dianteira");  
veiculo[1] = new Carro();  
veiculo[2] = new Moto();  
veiculo[3] = new Moto("XL125", 2019, 125);  
cout << endl << Veiculo::quantidade; //escreve a quantidade de veículos criados
```

E considere a seguinte função:

```
void escreve(Veiculo* veiculo){  
    veiculo->escreve();  
}
```

Sabe-se que, a bem da verdade, a função acima não poderá receber instâncias de veículos, uma vez que veículos não poderão ser instanciados diretamente, ao contrário, apenas especializados em carros ou motos. Além disso, a função *escreve()* deverá exibir na tela modelo, ano e tração, caso receba uma instância de carro; ou modelo, ano e cilindradas, caso receba uma instância de moto.

Implemente as três classes presentes neste enunciado, descrevendo-as:

- i. pelos atributos citados, observando as propriedades de encapsulamento e generalização; observar ainda as propriedades de membros estáticos;
 - ii. e **apenas** pelos métodos necessários para prover, corretamente, as funcionalidades acima – **não** implemente os demais – observe as propriedades da generalização e polimorfismo.
- a) (5,0) Classe Veículo
 - b) (5,0) Classe Carro
 - c) (5,0) Classe Moto

4. Principais objetos de avaliação:

As instruções abaixo criam instâncias de carro e moto sem tratar cenários em que uma exceção pode ser levantada por falta de recurso computacional para a criação dos respectivos objetos.

```
Veiculo* veiculo[4];
```

```
veiculo[0] = new Carro("Gol", 2020, "dianteira");
```

```
veiculo[1] = new Carro();
```

```
veiculo[2] = new Moto();
```

```
veiculo[3] = new Moto("XL125", 2019, 125);
```

(5,0) Reescreva as instruções apresentando um código que implemente o mecanismo de tratamento de exceção para este caso. Nele, quando a exceção ocorrer, além de comunica-la ao usuário do programa, o fluxo de execução deverá ser direcionado para a função *menu()*, cujo papel é prover um menu de opções para o usuário.



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**

Algoritmos e Estruturas de Dados I

Professor: Lúcio Mauro Pereira

Prova III – Valor: 35 pontos

13 de dezembro de 2023

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

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. Principais objetos de avaliação: **manipulação de arranjos e abordagens iterativa e recursiva.**

Construa uma função que verifique se é uma *string* um palíndromo ou não. Um palíndromo corresponde a uma palavra (ou frase, ou número) que possa ser lida de forma igual em ambas as formas, da esquerda para a direita e da direita para a esquerda. Por exemplo, é a palavra ARARA um palíndromo, tal como o é 1001. A função deverá gerar *verdadeiro*, caso a *string* enviada à função for um palíndromo, ou falso, caso contrário.

- c) $(5,0)$ Abordagem iterativa
- d) $(5,0)$ Abordagem recursiva

2. Principais objetos de avaliação: **manipulação de arranjos (matrizes)**

(5,0) Considere uma matriz quadrada de inteiros de dimensão $MAX_LIN \times MAX_LIN$. Implemente uma função inteira que receba tal matriz e retorne em qual posição da matriz quadrada está o maior valor, sendo: 0, caso o maior valor esteja na diagonal principal; +1, caso o maior valor esteja acima da diagonal principal; -1, caso esteja abaixo dela. Fique à vontade para distribuir a tarefa mais que uma função, caso julgue necessário.

3. Principal objeto de avaliação: **Princípios da Orientação por Objetos**

Considere uma Figura Geométrica que possa se especializar em retângulo ou em círculo.

Cor da figura, uma *string*, é uma propriedade comum a retângulo e círculo

Quando uma figura se especializa em retângulo, acrescenta-se os atributos *base* e *altura*, ambos expressos como números reais.

Quando uma figura se especializa em círculo, acrescenta-se o atributo *raio*, expresso em real.

Considere as seguintes instruções:

```
FiguraGeometrica* figuraGeometrica[4];
figuraGeometrica[0] = new Retangulo("Azul", 7, 3);
figuraGeometrica[1] = new Retangulo();
figuraGeometrica[2] = new Circulo();
figuraGeometrica[3] = new Circulo("Verde", 10);
cout << endl << FicuraGeometrica::quantidade; //escreve a quantidade de figuras criadas
```

E considere a seguinte função:

```
void escreve(FiguraGeometrica* figuraGeometrica){
    figuraGeometrica ->escreve();
}
```

Sabe-se que, a bem da verdade, a função acima não poderá receber instâncias de figuras geométricas, uma vez que figuras geométricas não poderão ser instanciadas diretamente, ao contrário, apenas especializadas em retângulos ou círculos. Além disso, a função escreve() deverá exibir na tela cor da figura e área da figura, sabendo que a área de um retângulo é obtida pelo produto da base e altura e a área de um círculo é obtida pelo produto do pi e o quadrado do raio (πr^2).

Implemente as três classes presentes neste enunciado, descrevendo-as:

- iii. pelos atributos citados, observando as propriedades de encapsulamento e generalização; observar ainda as propriedades de membros estáticos;
 - iv. e **apenas** pelos métodos necessários para prover, corretamente, as funcionalidades acima – **não** implemente os demais – observe as propriedades da generalização e polimorfismo.
- a) (5,0) Classe FiguraGeometrica
 - b) (5,0) Classe Retangulo
 - c) (5,0) Classe Circulo

4. Principais objetos de avaliação: **Tratamento de Exceção**

A instrução abaixo cria instâncias de Retangulo e Circulo sem tratar cenários em que uma exceção pode ser levantada por falta de recurso computacional para a criação dos respectivos objetos.

```
FiguraGeometrica* figuraGeometrica[4];
```

```
figuraGeometrica[0] = new Retangulo("Azul", 7, 3);
```

```
figuraGeometrica[1] = new Retangulo();
```

```
figuraGeometrica[2] = new Circulo();
```

```
figuraGeometrica[3] = new Circulo("Verde", 10);
```

(5,0) Reescreva as instruções apresentando um código que implemente o mecanismo de tratamento de exceção para este caso. Nele, quando a exceção ocorrer, o fluxo de execução deverá ser direcionado para a função *menu()*, cujo papel é prover um menu de opções para o usuário.