

# Prova 1

## Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

Nome: \_\_\_\_\_

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca **prova1.h**), função **main**, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- Todas as funções e procedimentos estarão automaticamente no módulo **prova1.h** e podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo **prova1.h**, considere que ela está implementada de forma correta.

Referências:

Função/Operador	Descrição	Biblioteca	Exemplo
%	retorna o resto da divisão	-	20 % 3 retorna 2

1. (4 points) Escreva uma **função** de nome **primosEntreSi** que verifica se dois números inteiros  $x$  e  $y$  são primos entre si. Se os números forem primos entre si a função deve retornar 1 e, caso negativo, deve retornar 0. Dois números são primos entre si se o máximo divisor comum entre eles for 1. Assim, a função **primosEntreSi** deve retornar 1 se os parâmetros de entrada forem 27 e 32 e 0 se os parâmetros de entrada forem 26 e 32. Essa função deve ter o seguinte protótipo:

```
int primosEntreSi(unsigned int x, unsigned int y);
```

2. (4 points) A função totiente, ou função phi ( $\phi$ ) – representada por  $\phi(x)$  – é, na teoria dos números, definida para um número natural  $x$  como sendo igual à quantidade de números menores ou igual a  $x$  co-primos com respeito a ele. Por exemplo,  $\phi(9) = 6$ , uma vez que 1, 2, 4, 5, 7 e 8 são co-primos de 9. Outros exemplos:  $\phi(1) = 1$  e  $\phi(15) = 8$ , pois 1, 2, 4, 7, 8, 11, 13 e 14 são co-primos de 15. Escreva uma função para calcular o totiente  $\phi(x)$  de um número inteiro e positivo  $x$ . Essa função deve ter o seguinte protótipo:

```
int phi(unsigned int x);
```

3. (4 points) Escreva um **procedimento** de nome **maiorPhi** que recebe como parâmetro dois endereços de memória de variáveis inteiras **end\_var1** e **end\_var2**. A função deve verificar qual dos inteiros armazenados em **end\_var1** e **end\_var2** tem o maior totiente. Depois, armazene esse inteiro em **end\_var1** e o seu totiente em **end\_var2**. Exemplo: se o valores armazenados em **end\_var1** e **end\_var2** forem 9 e 15, respectivamente, então o seu procedimento deve armazenar 15 em **end\_var1** e 8 em **end\_var2**.

4. (4 points) Complete o programa abaixo. Este programa deve ler dois inteiros do usuário e imprimir o inteiro com o maior totiente (ou phi) e o seu respectivo totiente (ou phi). Além disso, o programa deve pedir novos números inteiros para o usuário sempre que ele digitar algum inteiro menor ou igual a 0.

```
#include _____
#include <stdio.h>

void main() {

    int a, b;

    printf("\ndigite dois valores positivos: ");

    do {

        scanf("%d %d", &a, &b);

    } while(_____);

    _____;

    printf("\n o numero %d tem o maior phi: %d", _____);

}
```