

# Prova 1

## Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

1cm

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.
- As funções implementadas no módulo `prova1.h` 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.

**1. (2 points)**

Escreva um procedimento de protótipo `void imprimeAsteriscos(int n)` que imprime uma linha com  $n$  asteriscos. Exemplo: `imprimeAsteriscos(5)` imprime

```
*****
```

**2. (4 points)** Escreva um procedimento de protótipo `void imprimeTriangulo(int i, int h)` que imprime um triângulo lateral na tela formado por asteriscos. O triângulo deve começar imprimindo  $i$  asteriscos na primeira linha,  $i + 1$  na segunda, até  $h$  asteriscos na  $(h - i)$ ésima linha. Depois deve imprimir  $h - 1$  asteriscos,  $h - 2$ , até  $i$  asteriscos na última linha. Se  $i > 1$ , então é como se os lados do triângulo fossem cortados, os deixando com altura  $i$ . Para imprimir a linha com asteriscos, use a função `imprimeAsteriscos` do exercício anterior. Exemplo: `imprimeTriangulo(3,5)` imprime

```
***
****
*****
****
***
```

**3. (3 points)**

Escreva um programa que imprime na tela um triângulo lateral a partir da função `imprimeTriangulo` do exercício anterior. O seu programa deve pedir para o usuário os valores do tamanho do lado  $i$  e da altura  $h$  do triângulo. Antes de executar a função `imprimeTriangulo`, você deve garantir que  $i$  seja maior que zero e que  $h$  seja maior que  $i$ . Se isso não for satisfeito, você deve pedir novos valores para o usuário.

**4. (3 points)** Escreva um **procedimento** de nome `ordena` que recebe como parâmetro dois endereços de memória que armazenam pontos flutuantes (ponteiros para pontos flutuantes) `end_var1` e `end_var2`. A função deve acessar os valores armazenados nesses endereços e os ordenar em ordem crescente, de forma que o menor valor passe a ser armazenado em `end_var1` e o maior em `end_var2`.

**5. (4 points)** Complete o programa abaixo, que deve ler a altura de 10 casais (em metros, e.g. 1.85) e imprimir a altura da maior e da menor pessoa lida. O programa deve garantir que as alturas lidas sejam maiores que zero.

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

void main() {
    int count=0;
    float p1, p2, maior=_____, menor=_____;
    while(_____) {
        scanf("%f %f", &p1, &p2);
        if(_____) {
            ordena(_____);
            ordena(_____);
            ordena(_____);
            count++;
        }
        else
            printf("Erro! Digite novamente!\n");
    }
    printf("\nmaior: %f, menor: %f", maior, menor);
}
```