

# Prova 2

## Algoritmos e Estruturas de Dados I - turma E

**Professor:** Pedro O.S. Vaz de Melo

21 de maio de 2013 (valor: 20 pontos)

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

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

**Código de Honra para este exame (baseado no *Honor Code* da Universidade de Stanford):**

- 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:**

- Considere que todos os procedimentos e funções pedidas nesta prova serão implementados no módulo `prova2.h`.
- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo o módulo `prova2.h` quando necessário), 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.

**Referências:**

Função/Operador	Descrição	Biblioteca	Exemplo
<code>float log10(float x)</code>	retorna $\log_{10}x$	<code>math.h</code>	<code>log10(1000)</code> retorna 3.0
<code>float pow(float b, float e)</code>	retorna $b^e$	<code>math.h</code>	<code>pow(2,3)</code> retorna $2^3 = 8$
<code>%</code>	retorna o resto da divisão	-	<code>20 % 3</code> retorna 2

**1.** (10 points) O Comitê Olímpico Brasileiro quer um sistema para registrar o desempenho dos seus atletas. Assim, foi pedido para você desenvolver os seguintes itens, considerando a definição a seguir:

```
#define NUM_ATLETAS 193
```

**a.** (2 pts) Defina um novo tipo de dados chamado **Atleta** a partir de uma estrutura que tenha os seguintes campos: código do atleta (inteiro), número de medalhas de ouro, número de medalhas de prata, número de medalhas de bronze.

**b.** (2 pts) Implemente um procedimento de nome **preencheAtleta** que recebe um único **Atleta** como parâmetro e que preenche os seus campos com valores lidos do teclado. **A passagem de parâmetro deve ser feita por referência.**

**c.** (4 pts) Implemente uma função RECURSIVA de nome **numTotalMedalhasPorTipo** que recebe um vetor de atletas `atletas[]` e o tipo da medalha (`ouro=1`, `prata=2` ou `bronze=3`) e retorna o número total de medalhas desse tipo que os atletas brasileiros obtiveram. A função não deve fazer uso de variáveis globais, pode fazer uso da definição `NUM_ALUNOS`, e deve ter o seguinte protótipo.

```
int numTotalMedalhasPorTipo(Atleta atletas[], int tipoMedalha);
```

**d.** (2 pts) Escreva um **programa** para ler as informações de `NUM_ATLETAS` atletas do teclado e imprimir o número total de medalhas de ouro que eles obtiver. Para este programa você deve usar as funções **preencheAtleta** e **numTotalMedalhasPorTipo**, considerando que elas estão implementadas de forma correta.

**2. (6 points)** Para as questões a seguir, considere e faça uso, caso necessário, do trecho de código a seguir:

```
#define N 10

int random(int n) {
    return rand()%n;
}
```

**a. (2 pts)** Escreva um **procedimento** para preencher aleatoriamente uma matriz quadrada  $N \times N$  com 0s ou 1s. A probabilidade de preencher uma dada posição  $(i, j)$  com 0 deve ser a mesma de preencher com 1. Essa função deve ter o seguinte protótipo:

```
int preencheMatriz(int M[][N]);
```

**b. (2 pts)** Uma matriz identidade é uma matriz quadrada em que os elementos da diagonal principal têm valor um, e os demais elementos da matriz são zero. Escreva uma função que retorna 1 caso uma matriz quadrada  $N \times N$  seja uma matriz identidade e 0 caso contrário. Essa função deve ter o seguinte protótipo:

```
int verificaIdentidade(int M[][N]);
```

**c. (2 pts)** Escreva um programa para gerar uma matriz quadrada  $N \times N$  identidade usando somente a função **preencheMatriz** para popular a matriz. Esse programa pode usar a função **verificaIdentidade** e deve informar o número de vezes que a função **preencheMatriz** foi chamada até que a matriz tenha sido gerada de forma simétrica.

**3. (4 points)** Implemente um procedimento RECURSIVO que imprime o inverso de um número inteiro  $n$ . Exemplo: **imprimeInverso(4567)** imprime 7654 na tela. O procedimento não deve usar variáveis globais e deve ter o seguinte protótipo:

```
void imprimeInverso(int n);
```

**4. (2 points) (Questão bônus)** Implemente uma função RECURSIVA que retorna o inverso de um número inteiro  $n$ . Exemplo: **retornaInverso(4567)** retorna 7654. A função não deve usar variáveis globais e deve ter o seguinte protótipo:

```
int retornaInverso(int n);
```