

# Prova 1

## Algoritmos e Estruturas de Dados I - turma TM2

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

17 de setembro de 2013

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:**

- 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.

**Referências:**

Função/Operador	Descrição	Biblioteca	Exemplo
<code>float exp(float x)</code>	retorna $e^x$	<code>math.h</code>	<code>exp(1)</code> retorna $e^1 = 2.71828$
<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>int abs(int x)</code>	retorna $ x $	<code>stdlib.h</code>	<code>abs(-3)</code> retorna $ 3  = 3$
<code>double sin(double x)</code>	retorna $\text{seno}(x)$	<code>math.h</code>	<code>sin(3.1416)</code> retorna $\text{seno}(\pi) = 0$
<code>%</code>	retorna o resto da divisão	-	<code>20 % 3</code> retorna 2

**1. (5 points)** Um matemático maluco lhe procurou pois precisa de uma implementação em C da função:

$$f(x, \lambda) = \begin{cases} \lambda \text{seno}(-x) & \text{se } x < 0 \\ \lambda e^{-\lambda x} & \text{se } x \geq 0 \text{ e } x < |\lambda^3| \\ \lambda & \text{se } x \geq |\lambda^3| \end{cases} \quad (1)$$

Assim, implemente uma **função** de nome `flouca` que recebe os parâmetros  $x$  e  $\lambda$  e retorna  $f(x, \lambda)$ , todos pontos flutuantes.

**2. (5 points)** Escreva um **programa** que lê três números inteiros do teclado:  $x_{\min}$ ,  $x_{\max}$  e  $\lambda$  e imprime na tela os valores resultantes da função `flouca` com parâmetro  $\lambda$  para todos os valores inteiros entre  $x_{\min}$  e  $x_{\max}$  (inclusive para ambos). Enquanto o usuário inserir um valor de  $x_{\min}$  maior ou igual a  $x_{\max}$ , programa deve pedir a ele novos valores de  $x_{\min}$  e  $x_{\max}$ .

**3.** (5 points) Novamente, considere que a implementação desta questão será feita no módulo "prova1.h". Escreva uma **função** que retorna o mínimo múltiplo comum (MMC) entre dois números inteiros. O MMC de dois números  $x$  e  $y$  é o menor número inteiro que é divisível por  $x$  e também por  $y$ . Por exemplo, o MMC entre 16 e 12 é 48. Essa função deve ter o seguinte protótipo:

```
int mmc(int x, int y);
```

**4.** (5 points) Para as questões a seguir, considere que a implementação da letra **a** será feita no módulo "prova1.h". Para a letra **b**, considere que o módulo "prova1.h" tem a função **igualAoMenor** da letra **a** implementada corretamente.

**a.** (4 pts) Escreva um **procedimento** de nome **atribuiMMC** que recebe dois endereços de memória de variáveis inteiras **end\_var1** e **end\_var2**. O procedimento deve calcular o MMC entre os valores inteiros armazenados em **end\_var1** e **end\_var2** e deve, em seguida, armazenar o MMC calculado em ambos endereços. Assim, no final da execução, os dois endereços **end\_var1** e **end\_var2** devem conter o mesmo valor, que é o MMC entre os inteiros armazenados nesses endereços quando essa função foi chamada.

**b.** (1 pt) Complete o código abaixo:

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

void main(void) {
    int x,y;
    printf("Digite os valores de x e y\n");
    scanf(_____);
    atribuiMMC(_____);
    printf("Os novos valores de x e y sao: x=%d e y=%d\n", _____);
}
```