Prova 1

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

14 de abril de 2015

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.
- Quem entregar em até 1 hora ganha 10% da prova mais 0.5 pontos!

Referências:

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

1. (*8 points*) Para as questões a seguir, considere que as implementações serão feitas no módulo `"prova1.h"`.

a. (*4 pts*) Escreva uma **função** de nome `numDivsEmComum` que retorna o número de divisores em comum entre dois números inteiros x e y que são diferentes de 1. Assim, se os números forem primos entre si a função deve retornar 0. Por outro lado, a função `numDivsEmComum` deve retornar 5 se os parâmetros de entrada forem 40 e 20, pois são cinco os divisores em comum entre 20 e 40 que são diferentes de 1: $2, 4, 5, 10$ e 20. Essa função deve ter o seguinte protótipo:

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

b. (*4 pts*) Escreva um **programa** que lê dois números inteiros do teclado e imprime na tela o número de divisores em comum entre eles. Antes da impressão, você deve se certificar que ambos os números são maiores que 1 e distintos entre si, ou seja, o programa deve pedir os números ao usuário até que essas condições sejam satisfeitas.

2. (*4 points*) Escreva um **procedimento** de nome `troca3` que recebe como parâmetro dois endereços de memória que armazenam pontos flutuantes `end_var1` e `end_var2`. A função deve verificar se o conteúdo armazenado em `end_var2` é maior que o armazenado em `end_var1`. Caso positivo, você deve armazenar o conteúdo de `end_var2` em `end_var1` e o conteúdo de `end_var1` em `end_var2`.

3. (*3 points*)
Complete o código abaixo, considerando que as variáveis x e y vão ser usadas como parâmetros nas linhas 7, 8 e 9 (-1 ponto para cada resposta errada):

```
1: #include <stdio.h>
2: #include _____________________
3:
4: void main(void) {
5: float x,y;
6: printf("Digite os valores de x e y\n");
7: scanf(_______________________________);
8: troca3(__________________);
9: printf("O maior elemento digitado foi: _____", _____);
10:}
```