Prova 1

Algoritmos e Estruturas de Dados I - turma TE

Professor: Pedro O.S. Vaz de Melo

21 de março de 2014

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.

Referências:

Função/Operador	Descrição	Biblioteca	Exemplo
`float exp(float x)`	retorna e^x	`math.h`	`exp(1)` retorna $e^1 = 2.71828$
`%`	retorna o resto da divisão	-	`20 % 3` retorna 2

1. (*13 points*) Para as questões a seguir, considere que as implementações serão feitas no módulo `"prova1.h"`.

a. (*3 pts*) Um estatístico lhe procurou pois precisa de uma implementação em C da função densidade de probabilidade da distribuição exponencial, que é a seguinte:

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

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

b. (*4 pts*) Escreva uma **função** de nome `mdcEspecial` que retorna o máximo divisor comum (MDC) entre dois números inteiros x e y **se** o MDC for diferente de 1, de x e de y. Se o MDC for igual a 1, a x ou a y, a sua função deve retornar 0. Assim, a função `mdcEspecial` deve retornar 6 se os parâmetros de entrada forem 18 e 12 e 0 se os parâmetros de entrada forem 10 e 20. Essa função deve ter o seguinte protótipo:

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

c. (*2 pts*) Implementar uma função que recebe como parâmetro um número inteiro n e retorne 1 se ele for divisível por 2 ou por 7, mas não simultaneamente pelos dois, ou 0 caso contrário (divisível por 2 e por 7, ou por nenhum dos dois). Exemplo: essa função retorna 1 para o números 8 e 21 e retorna 0 para os números 14 e 15. Protótipo:

```
int ehDivisivelPor2ou7(int n);
```

d. (*4 pts*) Escreva um **procedimento** de nome `verificaMDC` que recebe como parâmetro dois endereços de memória de variáveis inteiras `end_var1` e `end_var2`. A função deve verificar se o MDC entre os valores x e y armazenados nesses endereços é igual a 1, x ou y. Se for igual, então você deve armazenar 0 em ambos endereços de memória, caso contrário, não precisa fazer nada.

2. (*5 points*) Escreva um programa para ler um número N do teclado que seja maior ou igual a 1. Dessa maneira, este programa deve pedir o número N do usuário até que o requisito de N ser maior ou igual a 1 seja atendido, ou seja, se o usuário der o valor de -8 à N, o programa deve pedir para ele um novo número. Depois disso, exiba a soma dos números inteiros menores que N que são divisíveis por 2 ou por 7, mas não por ambos.

3. (*2 points*) Complete o código abaixo, considerando que as variáveis x e y vão ser usadas como parâmetros nas linhas 7, 8 e 9:

```
1: #include <stdio.h>
2: #include ____________________
3:
4: void main(void) {
5: int x,y;
6: printf("Digite os valores de x e y\n");
7: scanf(_______________________________);
8: verificaMDC(___________________);
9: printf("Os novos valores de x e y sao: x=_____ e y=_____\n", _____, _____);
10: }
```