Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Programação de Computadores I – BCC701
Professor: David Menotti (menottid@gmail.com)

Lista de Exercícios 7 – Modularização

- ***Funções: Passagem de parâmetros.***

1) Escreva uma função que receba um número inteiro e imprima o mês correspondente ao número. Por exemplo, 2 corresponde à "fevereiro". O procedimento deve mostrar uma mensagem de erro caso o número recebido não faça sentido. Gere também um programa que leia um valor e chame o procedimento criado.

2) Escreva uma função que receba um número inteiro e o imprima na forma extensa. Por exemplo, para 1 a saída desejada é "Um". A função deve ser capaz de gerar o extenso dos números de 0 até 10, inclusive. Caso um número não compatível seja recebido o procedimento deve mostrar uma mensagem de erro. Crie também um programa que leia um valor inteiro e chame o procedimento criado acima para a impressão do número extenso.

3) Escreva uma função que gere um cabeçalho para um relatório. Esse procedimento deve receber um literal (*string*, ou cadeia de caracteres) como parâmetro. O cabeçalho tem a seguinte forma:

```
==========================================
 UFMG - Universidade Federal de Minas Gerais
 ICEx - Instituto de Ciências Exatas
 Disciplina de Programação de Computadores
 Nome: Fulano de Tal
==========================================
```

onde **`Fulano de Tal`**, corresponde ao parâmetro passado.

4) Escreva uma função que receba um número arábico inteiro e imprima o corresponde número em romano. Por exemplo, para 5 a saída desejada é "V". A função deve ser capaz de gerar o número romano para os 50 primeiros inteiros. Uma mensagem de erro deve ser mostrada caso um número fora dessa faixa seja recebido. Crie também um programa que leia um valor inteiro e chame o procedimento criado acima para a impressão do número romano.

5) Escreva uma função que receba um número natural e imprima os três primeiros caracteres do dia da semana correspondente ao número. Por exemplo, 7 corresponde à "SAB". O procedimento deve mostrar uma mensagem de erro caso o número recebido não corresponda à um dia da semana. Gere também um programa que utilize esse procedimento, chamando-o, mas antes lendo um valor para passagem de parâmetro.

- ***Funções que verificam uma situação, retorno booleano (verdadeiro, falso)***

6) Escreva uma função que receba um número inteiro. Esta função deve verificar se tal número é primo. No caso positivo, a função deve retornar 1, caso contrário zero. Escreva também um programa para testar tal função.

7) Escreva uma função que receba dois números inteiros ***x*** e ***y***. Essa função deve verificar se ***x*** é divisível por ***y***. No caso positivo, a função deve retornar 1, caso contrário zero. Escreva também um programa para testar tal função.

8) Um número é dito ser regular caso sua decomposição em fatores primos apresenta apenas potências de 2, 3 e 5. Faça uma função que verifique se um número é (retorne 1) ou não (retorne 0) regular. Escreva também um programa para testar tal função.

9) Criar uma função que determine se um caractere, recebido como parâmetro, é ou não uma letra do alfabeto. A função deve retornar 1 caso positivo e 0 em caso contrário. Escreva também um programa para testar tal função.

10) Um número é dito ser **capicua** quando lido da esquerda para a direita é o mesmo que quando lido da direita para a esquerda. O ano 2002, por exemplo, é **capicua**. Então, elabore uma função para verificar se um número possui essa característica. Caso o número seja **capicua**, a função deve retornar 1 e 0 em caso contrário. Escreva também um programa para testar tal função.

- ***Funções que retornam um valor calculado***

11) Criar uma função (não recursiva) que calcule e retorne o valor do fatorial de um número natural. A função deve retornar -1 caso não seja possível calcular o valor do fatorial. Escreva também um programa para testar tal função.

12) Criar uma função que calcule e retorne o número de arranjos de *n* elementos *p* a *p*. A fórmula do arranjo é a seguinte:

$$A_p^n = \frac{n!}{(n-p)!}$$

Caso não seja capaz de calcular tal arranjo a função deve retornar -1. Um programa para testar tal função também deve ser escrito.

13) Criar uma função que calcule e retorne o número de combinações de *n* elementos *p* a *p*. A fórmula de combinação é a seguinte:

$$C_p^n = \frac{n!}{p!(n-p)!}$$

Caso não seja capaz de calcular tal combinação a função deve retornar -1. Um programa para testar tal função também deve ser escrito.

14) Criar uma função que calcule e retorne o ***MAIOR*** entre dois valores recebidos como parâmetro. Um programa para testar tal função deve ser criado.

15) Criar uma função que verifique quantas vezes um número inteiro ***x*** é divisível por um número inteiro ***y***. A função deve retornar -1 caso não seja possível calcular. Escreva também um programa para testar tal função.

- ***Funções retornando mais de um parâmetro***

16) Construa uma função que efetue a ***TROCA*** dos valores de ***a*** por ***b,*** recebidos como parâmetro. Ou seja, essa função deve substituir o valor de ***a*** pelo de ***b***, e reciprocamente. Crie também um programa que leia dois valores quaisquer, e imprima os valores após a chamada da função ***TROCA***.

17) Construa uma função que receba três valores, ***a***, ***b*** e ***c***, retorne (passagem por referência) o ***MAIOR*** e o ***MENOR*** valor desses três. Deve ser criado um programa para utilizar tal função lendo os três valores e imprimindo o maior e o menor valor computado.

18) Construa uma função que receba dois valores inteiros ***a*** e ***b***, retorne (passagem por referência) o quociente, **div**, e o resto divisão, **mod**, de ***a*** por ***b***. A função deve retornar -1 caso não seja possível realizar as operações e 0 caso seja possível. Um programa para utilizar tal função deve ser criado, tratando o retorno da função.

19) Construa uma função que receba cinco valores e determine (retorne por passagem pro referência) o 2º e o 4º maior valores dentre eles. Construa também um programa para ler tais valores, e imprimir o resultado obtido com a chamada da função.

20) Construa uma função, que receba três coeficientes relativos à uma equação de segundo grau ($a.x^2 + b.x + c = 0$) e calcule suas raízes através da fórmula de báscara:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a} \qquad \Delta = b^2 - 4ac$$

A função deve levar em conta a possibilidade da existência de nenhuma, uma ou duas raízes. A função deve retornar o número de raízes ou -1 em caso de inconsistência. Os valores das raízes devem ser retornados. Construa também um programa para utilizar a função construída.

- ***Transformações***

21) Crie uma função que realize a conversão para Radianos (***rad***) a partir de Graus (***grad***), onde ***grad*** é passado como parâmetro e ***rad*** é retornado. Sabe-se que 180º (graus) está para π radianos. Crie também um programa para testar tal função.

22) Crie uma função que realize a conversão de Fahrenheit (***F***) para graus Celsius (***C***), onde ***F*** é passado como parâmetro e ***C*** é retornado. Sabe-se que os pontos de fusão e ebulição nas escalas Celsius e Fahrenheit são: 0ºC e 100ºC, e 32ºF e 212ºF, respectivamente. Crie também um programa para testar tal função.

23) Crie uma função que realize a conversão de Polegadas (***pol***) para Centímetros (***cm***), onde ***pol*** é passado como parâmetro e ***cm*** é retornado. Sabe-se que 1 polegada está para 2,54 centímetros. Crie também um programa para testar tal função.

24) Crie uma função que realize a conversão de pés (***feet***) para metros (***m***), onde ***feet*** é passado como parâmetro e ***m*** é retornado. Sabe-se que 1 metro está para 3,281 pés. Crie também um programa para testar tal função.

25) Crie uma função que realize a conversão da escala Kelvin (***K*** - escala absoluta) para a escala Fahrenheit (***F***). Sabe-se que 273K equivale a 32ºF e a cada variação de 10 unidades na escala Kelvin equivale a 18 na escala Fahrenheit. A função deve retornar zero caso não seja possível realizar a conversão e um em caso contrário. Crie também um programa para testar tal função.

- ***Funções recursivas***

26) Seja a série de Fibonacci:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

que pode ser definida recursivamente por:

$$Fib(n) = \begin{cases} 1 & \text{se} \quad n = 1 \vee n = 2 \\ Fib(n-1) + Fib(n-2) & \text{se} \quad n > 2 \end{cases}$$

Então escreva:

- Uma função recursiva que gere o termo de ordem n da série de Fibonacci.
- Um programa que, utilizando a função definida acima gere a série de Fibonacci até o termo de ordem 20.

27) Pode-se calcular o quociente da divisão, ***DIV***, de ***x*** por ***y***, dois números inteiros, usando-se a seguinte definição:

$$DIV(x,y) = \begin{cases} 1 + DIV(|x| - |y|, |y|), & \text{se} |x| > |y| \\ 0 & \text{se} |x| < |y| \\ 1 & \text{se} |x| = |y| \end{cases}$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. A função deve retornar -1 caso não seja possível realizar o cálculo. Além disso, crie um programa que leia os dois valores inteiros e utilize a função criada para calcular o quociente de ***x*** por ***y***, e imprima o valor computado.

28) Pode-se calcular o resto da divisão, ***MOD***, de ***x*** por ***y***, dois números inteiros, usando-se a seguinte definição:

$$MOD(x,y) = \begin{cases} MOD(|x| - |y|, |y|), & \text{se} |x| > |y| \\ |x| & \text{se} |x| < |y| \\ 0 & \text{se} |x| = |y| \end{cases}$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. A função deve retornar -1 caso não seja possível realizar o cálculo. Além disso, crie um programa que leia os dois valores inteiros e utilize a função criada para calcular o resto da divisão de ***x*** por ***y***, e imprima o valor computado.

29) O máximo divisor comum (***MDC***) de dois números inteiros ***x*** e ***y*** pode ser calculado usando-se uma definição recursiva:

$$MDC(x, y) = MDC(x - y, y), \text{ se } x > y .$$

Além disso, sabe-se que:

$$MDC(x, y) = MDC(y, x)$$

$$MDC(x, x) = x$$

Exemplo:

$$MDC(10,6) = MDC(4,6) = MDC(6,4) = MDC(2,4) = MDC(4,2) = MDC(2,2) = 2$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. Crie, também, um programa que leia os dois valores inteiros e utilize a função criada para calcular o ***MDC*** de ***x*** e ***y***, e imprima o valor computado.

30) O fatorial de um número ***n***, inteiro e positivo, pode ser definido recursivamente, ou seja:

$$n! = \begin{cases} 1 & se \quad n = 0 \\ n.(n-1)! & se \quad n \geq 1 \end{cases}$$

Então, pede-se que seja criada uma função recursiva que calcule o fatorial de um número n. A função deve retornar -1 caso não seja possível calcular o fatorial Além disso, crie um programa que leia um valor, utilize a função criada para calcular o fatorial e imprima o valor computado.