

## Funções e Procedimentos (Capítulo 5)

- 1) Marque verdadeiro (V) ou falso (F) nas afirmativas abaixo.  
Funções e procedimentos ...
  - ( ) permitem o reaproveitamento de código já implementado.
  - ( ) evitam que um trecho de código seja repetido várias vezes dentro de um mesmo programa.
  - ( ) facilitam a depuração de programas.
  - ( ) facilitam na compreensão do programa pois separam o programa em partes (blocos) que podem ser logicamente compreendidos de forma isolada.
- 2) Implemente uma função que recebe 2 números inteiros como parâmetro e retorna a multiplicação desses números. A multiplicação não deve ser feita utilizando o operador “\*”, em vez disso utilize apenas o operador de adição “+”. Implemente o programa principal para testar sua função.
- 3) Faça um programa que solicita ao usuário dois números inteiros diferentes “n1” e “n2” e chama uma função que calcula e retorna a soma de todos os números ímpares dentro do intervalo definidor por [n1,...,n2]. Antes de chamar a função, o programa principal deve testar se n1 é menor do que n2.
- 4) Implemente uma função que retorna o valor do seguinte somatório:

$$\sum_{i=3}^n (5*i+2)$$

O valor de “n” será passado como parâmetro para função.

- 5) Implemente uma função que retorna 1 quando o ano (passado como parâmetro) for bissexto e 0, caso contrário.  
  
Um ano não divisível por 100 e divisível por 4 é bissexto;  
Um ano divisível por 100 e divisível por 400 é bissexto;  
Os demais anos não são bissextos.
- 6) Escreva uma função que recebe como parâmetros 3 números inteiros correspondendo aos valores de dia, mês e ano, respectivamente, e retorna o número de dias já transcorridos neste ano. Utilize a função implementada no exercício anterior para determinar se o ano é bissexto (no ano bissexto o mês de fevereiro tem 29 dias).
- 7) Implemente a função “Primo” que recebe um número inteiro positivo como parâmetro e retorna 1 caso o número seja primo e 0, caso contrário. Por definição, um número é primo quando é divisível somente por si próprio e por 1.
- 8) Implemente um programa para calcular a soma dos N primeiros números primos, sendo N definido pelo usuário na função principal “main()”. O programa deverá ter as funções “Soma\_Primos” e “Primo”, sendo que a primeira será responsável pela soma dos números que forem primos e a segunda será responsável por verificar se o número em questão é primo ou não (mesma função que foi implementada no exercício anterior).
- 9) Implemente uma função que recebe um número inteiro como parâmetro e retorna o fatorial desse número. Se o número for menor do que zero, então a função deve retornar o valor -1. Implemente o programa principal para testar a função. Se o valor de retorno da função for “-1”, o programa principal deve informar que o número digitado está incorreto.
- 10) Implemente uma função que retorna o valor do seguinte somatório:

$$\sum_{i=3}^n i!$$

O valor de “n” será passado como parâmetro para função. Utilize a função fatorial implementada no exercício anterior.

- 11) Implemente uma função que calcula e retorna a potência de um número. A função deve receber como parâmetro o número (base) e o expoente. Faça o cálculo da potência utilizando o operador de multiplicação (\*).
- 12) Implemente o programa principal (*main*) e a função *float VAL (float x, int n, float t)* que recebe como parâmetro um valor *x*, uma taxa *t* e a quantidade de anos *n*. A função deve retornar o Valor Líquido Atual (*VAL*) calculado da seguinte maneira:

$$VAL = \frac{x}{(1+t)} + \frac{x}{(1+t)^2} + \frac{x}{(1+t)^3} + \dots + \frac{x}{(1+t)^n}$$

O programa principal deve chamar a função *VAL* passando parâmetros de teste (ou seja, valores para *x*, *n* e *t*) e, em seguida, exibir o resultado. A função *VAL* deve chamar a função que calcula potência, implementada no exercício anterior.

- 13) Implemente uma função que retorna o valor do seguinte somatório:

$$\sum_{i=3}^n i! / i^i$$

O valor de “n” será passado como parâmetro para função. Utilize as funções que calculam fatorial e potência, implementadas nos exercícios anteriores.

- 14) Qual o menor número inteiro entre 1 e 10000 que possui a maior quantidade de divisores? Quantos e quais são os divisores? Implemente um programa para encontrar a resposta.  
**Resposta: menor número 7560 com 64 divisores. Exiba os divisores**
- 15) Implemente uma **função** que calcula o mínimo múltiplo comum de dois número inteiros utilizando o método da decomposição simultânea dos números em fatores primos.

$$\begin{array}{r|l} 4, 15 & 2 \\ 2, 15 & 2 \\ 1, 15 & 3 \\ 1, 5 & 5 \\ 1, 1 & \end{array}$$

$$MMC = 2 * 2 * 3 * 5 = 60.$$

Implemente o programa principal para testar sua função.

- 16) Implemente uma função que recebe um número inteiro “num” como parâmetro e retorna o primeiro número primo maior do que “num”. Exemplo: ao receber 8, a função retornará 11; ao receber 17, a função retornará 19. Você deve implementar uma função específica que retorna 1 se o número for primo e 0, caso contrário.

- 17) Implemente um procedimento chamado “ calcular\_tempo” que recebe, por parâmetro, a hora de início e a hora de término de um jogo, ambas subdivididas em 2 valores distintos: horas e minutos. O procedimento deve exibir a duração do jogo em minutos, considerando que o tempo máximo de duração de um jogo é de 24 horas e que o jogo pode começar em um dia e terminar no outro.

Teste seu procedimento com os seguintes valores:

Início	Fim
00:00	23:59
23:50	00:50
10:50	10:51
10:40	10:39

- 18) Implemente uma função que recebe um número “n” como parâmetro e retorna o menor número inteiro positivo múltiplo de “n” que contém apenas dígitos menores ou iguais a 2. Exemplos de retorno da função f(n):  $f(2)=2$ ,  $f(3)=12$ ,  $f(7)=21$ ,  $f(42)=210$ ,  $f(89)=1121222$ . Implemente o programa principal para testar sua função.
- 19) Implemente um procedimento para calcular e exibir o seguinte somatório:  
 $S = f(1) + f(2) + f(3) + \dots + f(100)$ . A função f(n) foi definida no exercício anterior.  
**Resultado: 11363107**