

Lista de Exercícios 2 – Funções

- 1) Faça um programa que lê os três lados de um triângulo e determina o seu tipo, conforme códigos a seguir. Os códigos devem ser retornados por uma função de tipo *int*, que recebe os lados do triângulo como parâmetro. Protótipo da função:

```
int tipo_triangulo(float x, float y, float z);
```

O retorno da função deve ser conforme os códigos a seguir:

0. Os lados não formam um triângulo (ou seja, a soma de dois deles é menor ou igual ao outro lado);
 1. Triângulo equilátero;
 2. Triângulo isóceles;
 3. Triângulo escaleno.
- 2) Faça uma função que recebe três valores inteiros e retorna o maior valor. É preciso considerar que podem haver dois (ou mesmo os três) parâmetros iguais como sendo o maior valor. Por exemplo, os parâmetros poderiam ser 5, 8 e 8. Neste caso, a função deve retornar 8.
- 3) Escreva um programa que informa se um caractere digitado pelo usuário representa um dígito de 0 a 9. A verificação deve ser feita por uma função booleana (*int*) que recebe um *char* como parâmetro. Caso o caractere seja um dígito, converta-o para um valor inteiro e o armazene em uma variável *int*. Em seguida, mostre o valor inteiro na tela.
- 4) Faça uma função que recebe 2 parâmetros, *x* e *y*, e calcule a soma dos números ímpares entre eles (sem contar com eles mesmos). Repare que a função deve levar em conta de que *x* pode ser maior do que *y*. Por exemplo, para *x* = 6 e *y* = -5, temos a seguinte soma (em ordem crescente): $-3 + (-1) + 1 + 3 + 5 = 5$. Outro exemplo: para *x* = 3 e *y* = 10 temos $5 + 7 + 9 = 21$.
- 5) Faça um programa que, dados *k* e *n*, mostre na tela os *n* primeiros números primos acima de *k*. A verificação do número (se é ou não é primo) deve ser feita através de uma **função**.
- 6) Faça um programa que mostre na tela os *n* primeiros termos da sequência de Fibonacci. Por exemplo, dado *n* = 8, temos: 1, 1, 2, 3, 5, 8, 13 e 21. A determinação do *n*-ésimo termo da sequência deve ser feita por uma função **iterativa** que tem o seguinte protótipo:

```
int fibo(int n);
```

- 7) Escreva uma função que calcula o somatório dos *n* termos que são múltiplos de *k* a partir de *x*. Os parâmetros são determinados pelo usuário e a função é chamada pelo programa principal, que em seguida mostra o resultado na tela. Exemplo: para *n* = 3, *k* = 4 e *x* = 18, temos, $20 + 24 + 28 = 72$. Protótipo da função:

```
int soma_especial(int n, int k, int x);
```

- 8) Faça um programa que leia um inteiro **n** e determine a soma **S** da seguinte forma: $1 + 2 + 3 + 4 + \dots + n$. Escreva duas versões de funções: **iterativa** e **recursiva**.
- 9) Faça um programa que leia um inteiro **n** e utilize uma função (faça as versões **iterativa** e **recursiva**) para determinar a soma **S** da *série harmônica* definida a seguir:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

- 10) Faça um programa que leia um inteiro **n** e utilize uma função (faça as versões **iterativa** e **recursiva**) para calcular o somatório que determina o valor da constante **e**:

$$\sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!}$$