

# 4 Estruturas de Seleção

1. Usando o comando *if* sem a cláusula *else*, programe funções para receber um caractere e retornar verdadeiro ou falso. Cada função deve verificar se o caractere recebido por parâmetro é:

- a) uma vogal;
- b) uma letra minúscula;
- c) uma letra maiúscula;
- d) uma letra;
- e) uma consoante;
- f) um algarismo.

Em seguida, já podendo usar o *if* com *else*, chame os módulos programados a partir de um módulo *main* para testar seu programa.

2. A biblioteca *math.h* oferece a função *fabs*. Suponha que ela não exista e escreva um módulo para ter comportamento idêntico a esta função. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.
3. Programe uma função para receber em seus parâmetros dois números inteiros. Sua função deve indicar se o primeiro parâmetro é divisível pelo segundo. Em seguida, chame esta função a partir de um módulo *main* para testar seu programa. Neste programa, informe se um número fornecido pelo usuário é par ou ímpar.

4. Primeiramente, programe uma função para receber dois números e retornar o maior entre eles. Posteriormente, programe outra função para receber três números e retornar o maior entre eles. Tente escrever esta segunda função fazendo com que ela tenha uma única linha de código. Fique à vontade para chamar a primeira função para implementar a segunda. Em seguida, chame as funções criadas a partir de um módulo *main* para testar seu programa.
5. Para que três valores constituam os lados de um triângulo, é necessário que qualquer um dos lados seja inferior que a soma dos lados restantes. Programe um módulo para verificar se três valores são capazes de formar os lados de um triângulo. Dando continuidade, programe outra função. Esta segunda função deve receber três valores, e informar se tais medidas representam um triângulo equilátero (quando os três lados são iguais), um triângulo isósceles (quando dois dos três lados são iguais), um triângulo escaleno (quando os três lados são diferentes), ou se não constituem um triângulo. Esta segunda função pode retornar um inteiro entre 0 e 3, de acordo com a tabela abaixo:

Retorno	Tipo de triângulo
0	Não é triângulo
1	Escaleno
2	Isósceles
3	Equilátero

Em seguida, a partir de um módulo *main*, peça para o usuário fornecer três valores e faça seu programa responder a ele se ele forneceu valores incapazes de formar um triângulo, ou que tipo de triângulo é formado com os valores de entrada.

6. O Índice de Massa Corpórea (IMC) indica o grau de obesidade de um paciente. A tabela a seguir ilustra os diferentes níveis de obesidade em que uma pessoa pode se encontrar:

IMC (kg/m <sup>2</sup> )	Tipo de obesidade
< 25	Ausente
≥ 25 e < 30	Sobrepeso
≥ 30 e < 35	Obesidade Grau I
≥ 35 e < 40	Obesidade Grau II
≥ 40	Obesidade Mórbida

O Índice de Massa Corpórea é calculado pela expressão  $IMC = \frac{\text{peso}}{\text{altura}^2}$ .

Programe um módulo para receber em seus parâmetros o o peso e a altura de uma pessoa para retornar:

- 0, para ausente de obesidade;
- 1, para sobrepeso;
- 2, para obesidade grau 1;
- 3, para obesidade grau 2;
- 4, para obesidade mórbida.

Em seguida, a partir de um módulo *main*, peça para o usuário fornecer seu peso e sua altura para, em seguida, conhecer seu tipo de obesidade.

7. Programe uma função para receber um caractere e retornar a posição deste caractere no alfabeto, apenas se ele for uma letra. Você pode chamar uma das funções programadas no exercício 1 desta lista para facilitar sua programação. Porém, caso o caractere recebido no parâmetro desta função não seja uma letra, o retorno deste módulo deve ser o valor zero. Em seguida, chame a função criada a partir de um módulo *main* para testar seu programa.
8. Programe uma função para receber dois operandos numéricos e um operador. O operador é um caractere que indica qual operação deve ser feita com os primeiros parâmetros. Por exemplo, caracteres válidos para o operador podem ser: '+', '-', '\*', 'x', 'X', '/' ou ':'. Esta função deve retornar o resultado da operação em questão. Em seguida, chame a função criada a partir de um módulo *main* para testar seu programa.

9. Programe um módulo para receber um valor inteiro entre 1 e 12 e retornar o nome do mês equivalente. Se o parâmetro não for um inteiro válido para um mês do ano, o retorno da função deve ser um texto vazio. Em seguida, chame a função criada a partir de um módulo *main* para testar seu programa.
10. Implemente as funções do exercício 1 desta lista usando o operador ternário.
11. Re-escreva o programa abaixo para usar o operador || (OU lógico) de modo a simplificar o segundo comando IF:

```
#include <stdio.h>
main()
{
    char c;
    printf ("\nAperte uma tecla qualquer: ");
    c = getchar();
    if ((c >='a' ) && (c <= 'z')) c = c + 'A' - 'a';
    if (c == 'A')
        printf ("\nO caractere %c e' vogal", c);
    else if (c == 'E')
        printf ("\nO caractere %c e' vogal", c);
    else if (c == 'I')
        printf ("\nO caractere %c e' vogal", c);
    else if (c == 'O')
        printf ("\nO caractere %c e' vogal", c);
    else if (c == 'U')
        printf ("\nO caractere %c e' vogal", c);
    else printf ("\nO caractere %c nao e' vogal", c);
}
```

12. No Xintoquistão, a moeda corrente é o xinto. Neste país as cédulas existentes são as seguintes: X\$500,00, X\$100,00, X\$50,00, X\$10,00, X\$5,00 e X\$1,00. Escreva um programa que leia um inteiro N, calcule e imprima a quantidade mínima de cédulas de X\$ 500, X\$ 100, X\$ 50, X\$ 10, X\$ 5, X\$ 1 que são necessárias para pagar exatamente a importância dada pelo inteiro N. Por exemplo, se a importância a pagar for X\$ 309 então o programa deverá imprimir:

3 cédula(s) de X\$ 100  
1 cédula(s) de X\$ 5  
4 cédula(s) de X\$ 1

Nota: As cédulas com quantidade igual a zero não devem ser listadas.