

# Estrutura de decisão

## IF

A instrução `if` tem a função de provocar um desvio no programa baseado em uma condição. Pode ser representada da seguinte forma:

```
if (expressão)
    instrução;
```

ou, se estrutura composta,

```
if (expressão)
    instrução_1;
else
    instrução_2;
```

Pode ser facilmente transformada para o teste de múltiplas condições, como se vê no exemplo seguinte:

```
if (expressão)
    instrução_1;
else
    if (expressão)
        instrução_2;
    else
        instrução_3;
```

Outro exemplo:

```
main()
{
    int x, y, z;

    ...
    if (x >= y) {
        z = x;
        ...
    }
    else {
        z = y;
        ...
    }
    ...
}
```

Notar a colocação das chaves, definindo um bloco de comandos para a estrutura de decisão.

## O operador ?: (ternário)

O operador `?:` (condição ternária) é, como o nome indica, um operador que necessita de três operandos, que podem ser resultados de expressões. Em certas situações pode substituir a instrução `if`. Tem a forma:

```
expressão_1 ? expressão_2 : expressão_3;
```

Tem como resultado o valor de `expressão_2` ou de `expressão_3` conforme o valor de `expressão_1`, se for verdadeiro (`!= 0`) ou falso (`== 0`) respectivamente. Por exemplo, para colocar o máximo de dois valores numa variável podemos simplesmente escrever:

```
z = (a > b) ? a : b;
```

que é perfeitamente equivalente a:

```
if (a > b)
    z = a;
else
    z = b;
```

## A instrução `switch`

A instrução `switch` permite uma escolha múltipla de execução baseada no valor de uma expressão. A sua estrutura geral é a seguinte:

```
switch (expressão) {
    case item_1:
        instrução_1;
        break;
    case item_2:
        instrução_2;
        break;
    ...
    case item_n:
        instrução_n;
        break;
    default:
        instrução;
}
```

Os `items` que aparecem a seguir à palavra `case` são valores constantes (não podem ser expressões nem variáveis). Se a `expressão` inicial coincidir com algum desses items executa-se a instrução que vem logo a seguir. Caso contrário, e se a palavra `default` estiver presente (não é obrigatório), passa-se a essa instrução; se não estiver presente passa-se à instrução que se segue à instrução `switch`. A palavra `break` (é uma instrução do C) faz com que se passe imediatamente à instrução que se segue à presente (a instrução que virá a seguir a esta instrução `switch` no programa). Se não estiver presente a execução continua para a instrução do próximo `case`. Podemos considerar o seguinte exemplo:

```
switch (letter) {
    case 'a':
        nr_de_vogais++;
        break;
    case 'e':
        nr_de_vogais++;
        break;
    case 'i':
        nr_de_vogais++;
        break;
    case 'o':
        nr_de_vogais++;
        break;
    case 'u':
        nr_de_vogais++;
        break;
    nr_de_vogais++;
}
```

```
    break;
case ' ':
    nr_de_espacos++;
    break;
default:
    nr_de_consoantes++;
}
```

## Exercícios

1. Leia dois números reais. Se o primeiro for maior escreva "Maior"; se for menor escreva "Menor" e se forem iguais escreva "Iguais".
2. Leia um valor inteiro entre 1 e 12 e assumo que esse valor representa um mês. Escreva o seu nome. Se for lido um valor que não corresponda a um mês, mostrar uma mensagem de erro.
3. Leia três inteiros representando uma data (dia, mês e ano). Escreva, da mesma forma, a data do dia seguinte.
4. Faça um programa que receba três notas de um aluno, de no máximo 10 pontos. Calcule a média aritmética e mostre sua situação de acordo com a tabela abaixo. Para alunos de exame, calcule a nota que ele deverá tirar para ser aprovado, considerando a nota para ser aprovado em exame 5,0 numa nova média calculada pela Média do aluno + Prova de 10 pontos.

De	Até	Situação
0,0	3,0	Reprovado
3,0	7,0	Exame
7,0	10,0	Aprovado