

Linguagem de Programação C

Comandos de Seleção e Desvio

Cristiano Lehrer

<http://www.ybadoo.com.br/>

Valores Lógicos

- Em C não existe nenhum tipo específico de dados para armazenar valores lógicos.
- Em C o valor lógico FALSO é representado por 0 (ZERO).
- Tudo aquilo que seja diferente de 0 (ZERO) representa o valor lógico VERDADE:
 - O valor lógico VERDADE em C não é o valor 1 (UM), mas sim qualquer valor diferente de 0 (ZERO).
 - O valor 1 (UM) é apenas um dos valores possíveis para representar VERDADE.

Operadores Relacionais (1/2)

Operador	Nome	Exemplo	Significado do Exemplo
==	igualdade	$a == b$	a é igual a b?
>	maior que	$a > b$	a é maior que b?
>=	maior ou igual que	$a >= b$	a é maior ou igual a b?
<	menor que	$a < b$	a é menor que b?
<=	menor ou igual que	$a <= b$	a é menor ou igual a b?
!=	diferente de	$a != b$	a é diferente de b?

Uma expressão que contenha um operador relacional devolve sempre como resultado o valor lógico VERDADE (1) ou FALSO (0).

Operadores Relacionais (2/2)

```
#include <stdio.h>
main()
{
    int x, y;

    printf("Introduza dois inteiros: ");
    scanf("%d%d", &x, &y);
    printf("O resultado de %d == %d : %d\n", x, y, x == y);
    printf("O resultado de %d > %d : %d\n", x, y, x > y);
    printf("O resultado de %d >= %d : %d\n", x, y, x >= y);
    printf("O resultado de %d < %d : %d\n", x, y, x < y);
    printf("O resultado de %d <= %d : %d\n", x, y, x <= y);
    printf("O resultado de %d != %d : %d\n", x, y, x != y);
}
```

if-else (1/2)

```
if (condição)
    instrução1;
[else instrução2;]
```

- Funcionamento:
 - A condição é avaliada.
 - Se o resultado da condição for verdadeiro, executa a instrução1.
 - Se o resultado da condição for falso, executa a instrução2 (caso exista o **else**).

if-else (2/2)

```
#include <stdio.h>
main()
{
    int x;

    printf("Introduza um número inteiro: ");
    scanf("%d", &x);

    if(x >= 0)
        printf("Número positivo\n");
    else
        printf("Número negativo\n");
}
```

Operadores Lógicos (1/2)

Operador	Significado	Exemplo
&&	AND (E lógico)	<code>x >= 1 && x <= 19</code>
	OR (OU lógico)	<code>x == 1 x == 2</code>
!	NOT (negação lógica)	<code>!continuar</code>

Operadores Lógicos (2/2)

```
#include <stdio.h>
main()
{
    float salario;
    char est_civil;

    printf("Qual o salário: ");
    scanf("%f", &salario);
    printf("Qual o estado civil: ");
    scanf(" %c", &est_civil);
    if(est_civil == 'C' || est_civil == 'c')
        printf("Imposto: %.2f\n", salario * 0.09);
    else
        if(est_civil == 'S' || est_civil == 's')
            printf("Imposto: %.2f\n", salario * 0.10);
        else
            printf("Estado civil incorreto!\n");
}
```


Operador Condicional ? (1/2)

```
condição ? expressão1 : expressão2;
```

- Funcionamento:
 - A condição é avaliada.
 - Se o resultado for verdade, o resultado de toda a expressão é o valor devolvido por expressão1.
 - Se o resultado for falso, o resultado de toda a expressão é o valor devolvido por expressão2.

Operador Condicional ? (2/2)

```
#include <stdio.h>

main()
{
    float salario;

    printf("Qual o salário: ");
    scanf("%f", &salario);
    salario = salario > 1000.0 ? salario * 1.05 : salario * 1.07;
    printf("Novo salário: %.2f\n", salario);
}
```

switch (1/5)

switch (expressão)

```
{  
    case constante: instruções1;  
    case constante: instruções2;  
    ...  
    case constante: instruçõesn;  
    [default: instruções]  
}
```

switch (2/5)

- Funcionamento:
 - Se o valor da expressão for igual a alguma das constantes que seguem os vários `case`, então são executadas as instruções que seguem o `case` correspondente.
 - Se o valor da expressão não for igual a nenhuma das constantes apresentadas pelos `case`, então são executadas as instruções que seguem o `default`.
 - Como se pode observar pela sintaxe, o `default` é opcional. No caso de o valor da expressão não ser igual a nenhum dos `case`, nada é executado, terminando o `switch`. O programa continua na instrução seguinte ao `switch`.

switch (3/5)

```
#include <stdio.h>
main()
{
    char est_civil;

    printf("Qual o estado civil: ");
    scanf("%c", &est_civil);
    switch(est_civil)
    {
        case 'C': printf("Casado\n");
        case 'S': printf("Solteiro\n");
        case 'D': printf("Divorciado\n");
        case 'V': printf("Viúvo\n");
        default : printf("Estado civil incorreto!\n");
    }
}
```

switch (4/5)

```
#include <stdio.h>
main()
{
    char est_civil;

    printf("Qual o estado civil: ");
    scanf("%c", &est_civil);
    switch(est_civil)
    {
        case 'C': printf("Casado\n"); break;
        case 'S': printf("Solteiro\n"); break;
        case 'D': printf("Divorciado\n"); break;
        case 'V': printf("Viúvo\n"); break;
        default : printf("Estado civil incorreto!\n");
    }
}
```

switch (5/5)

```
#include <stdio.h>
main()
{
    char est_civil;

    printf("Qual o estado civil: ");
    scanf("%c", &est_civil);
    switch(est_civil)
    {
        case 'c':
        case 'C': printf("Casado\n"); break;
        case 's':
        case 'S': printf("Solteiro\n"); break;
        case 'd':
        case 'D': printf("Divorciado\n"); break;
        case 'v':
        case 'V': printf("Viúvo\n"); break;
        default : printf("Estado civil incorreto!\n");
    }
}
```