Aula 05

MC 102 - Algoritmos e Programação de Computadores

Expressões Relacionais, Expressões Lógicas e Comandos Condicionais.

Expressões Relacionais

São aquelas que realizam uma comparação entre duas expressões e retornam **verdadeiro** ou **falso**

<expressão_1> <op_relacional> <expressão_2>

Exemplo:

tempo de forno == 40 min?

Operadores Relacionais

<op_relacional> (operadores relacionais) pode ser:

- Igual a (igualdade) ==
- Maior ou igual a >=
- Maior que >
- Menor ou igual a <=
- Menor que <
- Diferente de !=

Por que == e não = ?

A linguagem C usa o = como operador de atribuição;

O operador relacional de igualdade é ==

- Se uma variável inteira A tem conteúdo 5, então:
  - A == 6 retorna "falso"
  - A = 6 atribui 6 em A, que deixa de valer 5.

Verdadeiro e Falso

Como “verdadeiro”e “falso” são definidos na linguagem C?

- Falso
  - Definido como 0 (Zero)
- Verdadeiro
  - Definido como 1 (um) ou qualquer outro valor diferente de 0 para o resultado de uma expressão.

Verdadeiro e Falso

Para não esquecer...

NÃ0 p/ falso

S1M p/ verdadeiro

Exemplos

Expressão (a=5 b=6)	Resultado
• a == b	• 0 (falso)
• a != b	• 1 (verdadeiro)
• a >= b	• 0 (falso)
• a <= b	• 1 (verdadeiro)
• a > b	• 0 (falso)
• a < b	• 1 (verdadeiro)
• a = b (o que acontece?)	• 6 (verdadeiro) – um erro comum

Expressões Lógicas

São aquelas que realizam uma operação lógica e retornam verdadeiro ou falso

<expressão_1> <op_lógico> <expressão_2>

onde:

<expressão_1> e <expressão_2> são expressões relacionais

Exemplo:

Tempo de forno == 40 min e situação do bolo == assado?

Operadores Lógicos

<op_relacional> (operadores lógicos)

&&	E Lógico (AND)
||	OU Lógico (OU)
!	Não Lógico (NOT)

&& (AND) - Tabela Verdade

Retorna 1 (verdadeiro) **apenas** quando **ambas** as expressões são **verdadeiras**. Logo:

expr1	expr2	expr1 && expr2
0 (F)	0 (F)	0 (Falso)
0 (F)	1 (V)	0 (Falso)
1 (V)	0 (F)	0 (Falso)
1 (V)	1 (V)	1 (Verdadeiro)

Ex: (a >= 0) && (b >= 0)

|| (OR) - Tabela Verdade

Retorna 1 (verdadeiro) quando **pelo menos uma** das expressões são **verdadeiras**. Logo:

expr1	expr2	expr1 || expr2
0 (F)	0 (F)	0 (Falso)
0 (F)	1 (V)	1 (Verdadeiro)
1 (V)	0 (F)	1 (Verdadeiro)
1 (V)	1 (V)	1 (Verdadeiro)

Ex: (a >= 0) || (b >= 0)

! (NOT) - Tabela Verdade

Retorna 1 (verdadeiro) quando a expressão é **falsa** e vice-versa. Logo:

expr1	!expr1
0 (F)	1 (Verdadeiro)
1 (V)	0 (Falso)

Ex: ! (a == 0)

Equivalências

Expressões lógicas equivalentes:

!(a == b)	a != b
!(a != b)	a == b
!(a > b)	a <= b
!(a < b)	a >= b
!(a >= b)	a < b
!(a <= b)	a > b

Equivalências – Leis de Morgan

- ! (a || b):

a	b	a||b	!(a||b)	!a	!b	!a&&!b
V	V	V	F	F	F	F
V	F	V	F	F	V	F
F	V	V	F	V	F	F
F	F	F	V	V	V	V

- ! (a && b):

a	b	a&&b	!(a&&b)	!a	!b	!a||!b
V	V	V	F	F	F	F
V	F	F	V	F	V	V
F	V	F	V	V	F	V
F	F	F	V	V	V	V

Precedência dos Operadores

parênteses
++ , -- , !
* , /
%
+ , -
< , <= , > , >=
== , !=
&&
||
= , += , -= , /= , *=

Dica:

- Não confie cegamente na precedência dos operadores, você pode ser traído!
- Sempre use parênteses para garantir a precedência, principalmente em expressões lógicas!

Comando Condicionais

Falamos de expressões relacionais e condicionais, mas como verificar se uma condição é verdadeira ou falsa?

Comando Condicionais

...

Se o forno está a 180ºC **então**

Coloque o bolo no forno

Senão espere até 180ºC

Se tempo de forno é 40 min **então**

Se bolo está assado **então**

Tire do forno. Bolo está pronto

Senão espere mais 1 min para o novo teste

Senão espere até 40 min

Comando Condicionais

Falamos de expressões relacionais e condicionais, mas como verificar se uma condição é verdadeira ou falsa?

Ex: algoritmo para o maior entre 2 inteiros

Informe a
Informe b
Se a > b, então
    a é o maior
Senão, se b > a, então
    b é o maior
Senão
    a e b são iguais.

Comando if

Testa se uma condição é verdadeira. Sendo verdadeira, executa o comando seguinte.

```
if ( expr_lógica )
    comando;
```

Ex:
```
if( a == 0 )
    printf("a não pode ser zero!\n");
```

Importante!

Nunca use **;** após a definição de um **if**.

~~`if( a == 0 );`~~

Comando if - else

Variação do **if**. Define comando para ser executado caso a expressão lógica seja verdadeira e comando para o caso dela ser falsa (**else**)

```
if( expr_lógica )
    comando;
else
    comando;
```

Ex:

```
if( a < 0 )
   printf("a menor que zero!\n");
else
   printf("a maior ou igual a zero!\n");
```

Ifs aninhados

Às vezes, precisamos aninhar (cascatear) mais de um if para testar algumas condições.

...
Se tempo de forno é 40 min **então**
    Se bolo está assado **então**
        Tire do forno. Bolo está pronto
    Senão espere mais 1 min para o novo teste
Senão espere até 40 min

Ifs aninhados

Quando os comandos serão executados?

```
if( expr_lógica_1 )
  if ( expr_lógica_2)
    comando_1;
  else
    comando_2;
```

Ifs aninhados

Quando os comandos serão executados?

```
if( expr_lógica_1 )
  if ( expr_lógica_2)
    comando_1;
  else
    comando_2;
```

```
if( expr_lógica_1 )
  if ( expr_lógica_2)
    comando_1;
  else
    comando_2;
```

O **else** está associado ao último **if**, de **expr_lógica_2**

If para vários comandos

Outras vezes, uma condição está associada à execução de vários comandos.

...

Se bolo está assado **então**

Tire do forno

Bolo está pronto

Senão ...

If para vários comandos

Quando os comandos serão executados?

```
if( expr_lógica_1 )
  comando_1;
  comando_2;
  comando_3;
```

?

If para vários comandos

Quando os comandos serão executados?

```
if( expr_lógica_1 )
  comando_1;
  comando_2;
  comando_3;
```

```
if( expr_lógica_1 )
  comando_1;
  comando_2;
  comando_3;
```

comando_2 e **comando_3** serão sempre executados

Blocos de Comandos

Define um conjunto de comandos para serem avaliados em grupo. Para isso usam-se chaves ({ e })

```
If ( expr_lógica_1 ) {
   if ( expr_lógica_2)
      comando_1;
} else
   comando_2;
```

```
If ( expr_lógica_1 ) {
   comando_1;
   comando_2;
   comando_3;
}
```

Exemplo 1 - Algoritmo

Maior entre 2 números inteiros

Entrada de dados:

2 inteiros a e b

Saída

qual dentre eles é o maior

Algoritmo

Informe a
Informe b
Se a > b, então
a é o maior
Senão, se b > a, então
b é o maior
Senão
a e b são iguais.

Exemplo 1 - Programa

Maior entre 2 números inteiros

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a,b;

    printf("Informe a e b inteiros: ");
    scanf("%d %d", &a, &b);
    if (a > b)
      printf("\n%d é maior que %d\n", a, b);
    else {
      if (a < b)
        printf("\n%d é maior que %d\n", b, a);
      else
        printf("\na e b são iguais\n");
    }
}
```

Exercício

Faça um programa que verifique se um número inteiro qualquer é par ou ímpar.

Exercício - Algoritmo

Verificar se número é par ou ímpar

Entrada de dados:

um número inteiro n

Saída

informação se o número é par ou ímpar

Algoritmo

Informe o número inteiro n
Se (n / 2 tem resto 0)
 o número é um número par
Senão
 o número é um número ímpar

Exercício- Solução

Verifica se um número é par ou ímpar

```
#include <stdio.h>
int main(int argc, char **argv) {
    int n;

    printf("Digite um número inteiro: ");
    scanf("%d", &n);
    if ((n % 2) == 0)
      printf("\nO número %d é um número par\n", n);
    else
      printf("\nO número %d é um número ímpar\n", n);
}
```

Exercício

Escreva um programa que, dado o comprimento de três segmentos de reta, determine se eles formam um triângulo e, caso formem, diga se o triângulo é equilátero, isósceles ou escaleno.