

MC-102 — Aula 04

Expressões Relacionais, Lógicas e Comandos Condicionais

Instituto de Computação – Unicamp

29 de Fevereiro de 2016

Roteiro

- 1 Expressões relacionais
- 2 Expressões lógicas
- 3 Comandos condicionais
- 4 Exercícios

Expressão

- Já vimos que constantes e variáveis são expressões.

Exemplo

```
a = 10;  
a = b;
```

- Vimos também que operações aritméticas também são expressões.

Exemplo

```
a = 2 + 2;  
a = 10 / 3;  
a = a + 1;
```

Expressões relacionais

Expressões relacionais são aquelas que realizam uma **comparação** entre duas expressões e retornam

- 1 **Zero (0)**, se o resultado é falso
- 2 **Um (1)**, ou qualquer outro número diferente de zero, se o resultado é verdadeiro.

Operadores Relacionais

Os operadores relacionais da linguagem C são:

- $==$: igualdade.
- $!=$: diferente.
- $>$: maior que.
- $<$: menor que.
- $>=$: maior ou igual que.
- $<=$: menor ou igual que.

Expressões relacionais

- *expressao* == *expressao*: Retorna verdadeiro quando as expressões forem iguais.

```
9 == 9 //Resultado é 1, True
```

```
9 == 10 //Resultado é 0, False
```

- *expressao* != *expressao*: Retorna verdadeiro quando as expressões forem diferentes.

```
9 != 9 //Resultado é 0, False
```

```
9 != 10 //Resultado é 1, True
```

Expressões relacionais

- *expressao* > *expressao*: Retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.

9 > 5 //Resultado é 1, True

- *expressao* < *expressao*: Retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.

9 < 5 //Resultado é 0, False

Expressões relacionais

- *expressao* \geq *expressao*: Retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.

```
9 >= 5 //Resultado é 1, True
```

- *expressao* \leq *expressao*: Retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.

```
9 <= 5 //Resultado é 0, False
```


O que será impresso pelo programa?

```
#include <stdio.h>

int main(){
    int a, b;

    printf("%d\n", 9 > 3);

    printf("%d\n", (3*4)/2 != (2*3) );

    a = 1;
    b = -1;
    printf("%d\n", a!=b);
}
```

Expressões lógicas

Expressões lógicas são aquelas que realizam uma operação lógica (**ou**, **e**, **não**, etc...) e retornam verdadeiro ou falso (como as expressões relacionais).

Operadores Lógicos

Na linguagem C temos os seguintes operadores lógicos:

- `&&`: operador E.
- `||`: operador OU.
- `!`: operador NÃO.

Expressões lógicas

- *expressao* && *expressao*: Retorna verdadeiro quando ambas as expressões são verdadeiras. Sua tabela verdade é:

<i>Op₁</i>	<i>Op₂</i>	<i>Ret</i>
V	V	V
V	F	F
F	V	F
F	F	F

Qual o resultado da expressão lógica abaixo?

```
a = 0;
```

```
b = 0;
```

```
(a == 0 && b == 0)
```

Expressões lógicas

- *expressao* || *expressao*: Retorna verdadeiro quando pelo menos uma das expressões é verdadeira. Sua tabela verdade é:

<i>Op₁</i>	<i>Op₂</i>	<i>Ret</i>
V	V	V
V	F	V
F	V	V
F	F	F

Qual o resultado da expressão lógica abaixo?

a = 0;

b = 1;

(a == 0 || b == 0)

Expressões lógicas

- **!** *expressao*: Retorna verdadeiro quando a expressão é falsa e vice-versa. Sua tabela verdade é:

Op_1	Ret
V	F
F	V

Qual o resultado da expressão lógica abaixo?

$a = 0;$

$b = 1;$

$!(a != b)$

O que será impresso pelo programa?

```
#include <stdio.h>
```

```
int main(){
```

```
    printf("%d\n", (8>9) && (10!=2));
```

```
    printf("%d\n", (14 > 100) || (2>1));
```

```
    printf("%d\n", ( !(14>100) && !(1>2) ) );
```

```
}
```

Comandos condicionais

Um comando condicional é aquele que permite decidir se um determinado bloco de comandos deve ou não ser executado, a partir do resultado de uma expressão relacional ou lógica.



Bloco de comandos

- É um conjunto de instruções agrupadas.
- Limitada pelos caracteres { e }.

Exemplo

```
int main()  
{           ← Início do bloco de comandos  
  int a;  
  a=1;  
}           ← Fim do bloco de comandos
```

Comandos condicionais

- O principal comando condicional da linguagem C é o **if**, cuja sintaxe é:

```
if(expressão relacional ou lógica)
    um único comando;
```

ou também

```
if (expressão relacional ou lógica) {
    comando 1;
    comando 2;
    ...
    comando n;
}
```

- Os comandos são executados somente se a expressão relacional/lógica for verdadeira.

Comandos condicionais

O programa abaixo determina se um valor é ímpar.

```
#include <stdio.h>

int main () {
    int a;
    scanf("%d", &a);
    if ((a % 2) != 0) {
        printf ("O valor é ímpar.\n");
    }
}
```

Comandos condicionais

Lembrando como C representa os valores Falso e Verdadeiro, o programa pode ser alterado para:

```
#include <stdio.h>

int main () {
    int a;
    scanf("%d", &a);
    if (a % 2) {
        printf ("O valor é ímpar.\n");
    }
}
```

Comandos condicionais

- Uma variação do comando **if** é o **if/else**, cuja sintaxe é:

```
if (expressão relacional ou lógica){
    comandos executados se a expressão é verdadeira;
} else {
    comandos executados se a expressão é falsa;
}
```

Comandos condicionais

Exemplo: Determinando o menor de dois números:

```
int main(void){
    int a,b;

    scanf("%d", &a);
    scanf("%d", &b);

    if(a < b){
        printf("O menor numero e: %d\n", a);
    }else{
        printf("O menor numero e:%d\n",b);
    }

}
```

Comandos condicionais

- Note que o **if** é um comando, e como tal pode aparecer dentro do bloco de comandos de outro **if**.

Exemplo: Usando apenas operadores relacionais, vamos escrever um programa que lê um número e verifica em qual dos seguintes casos o número se enquadra:

- Par e menor que 100.
- Par e maior ou igual a 100.
- Ímpar e menor que 100.
- Ímpar e maior ou igual a 100.

Comandos condicionais

```
#include <stdio.h>

int main(){
    int a;
    printf("Digite um número inteiro:");
    scanf("%d", &a);
    if( a % 2 == 0){
        if( a < 100)
            printf("0 número é par e menor que 100\n");
        else
            printf("0 número é par e maior ou igual a 100\n");
    }else{
        if ( a < 100)
            printf("0 número é ímpar e menor que 100\n");
        else
            printf("0 número é ímpar e maior que 100\n");
    }
}
```

Se você pudesse usar operadores lógicos, como você poderia refazer este programa?

Comandos condicionais

```
#include <stdio.h>

int main(){
    int a;
    printf("Digite um número inteiro:");
    scanf("%d", &a);
    if( ( a % 2 == 0) && (a<100) )
        printf("O número é par e menor que 100\n");
    if( ( a % 2 == 0) && (a>=100) )
        printf("O número é par e maior ou igual a 100\n");
    if( ( a % 2 != 0) && (a<100) )
        printf("O número é ímpar e menor que 100\n");
    if( ( a % 2 != 0) && (a>=100) )
        printf("O número é ímpar e maior que 100\n");
}
```

Comandos condicionais

```
if (cond1) {  
    if (cond2)  
        comando1;  
} else  
    comando2;
```

Quando o **comando2** é executado?

Comandos condicionais

```
if (cond1) {  
    if (cond2)  
        comando1;  
} else  
    comando2;
```

Quando o **comando2** é executado?

Resposta: quando cond1 for falsa.

Comandos condicionais

```
if (cond1){  
    if (cond2)  
        comando1;  
    else  
        comando2;  
}else{  
    if(cond3)  
        comando3;  
    else  
        comando4;  
}
```

Quando o **comando4** é executado?

Comandos condicionais

```
if (cond1){
    if (cond2)
        comando1;
    else
        comando2;
}else{
    if(cond3)
        comando3;
    else
        comando4;
}
```

Quando o **comando4** é executado?

Resposta: quando a **cond1** for falsa e **cond3** for falsa.

Comandos condicionais

Use chaves e indentação para deixar claro a qual comando condicional um outro comando pertence!!

```
if (cond1)
if (cond2)
    comando1;
else
    comando2;
```

Quando o **comando2** é executado?

Comandos condicionais

Use chaves e indentação para deixar claro a qual comando condicional um outro comando pertence!!

```
if (cond1)
if (cond2)
    comando1;
else
    comando2;
```

Quando o **comando2** é executado?

Resposta: O comando **if-else** é um único comando, portanto ele está dentro do primeiro **if**. Logo comando2 é executado quando cond1 for verdadeira e cond2 falsa.

Comandos condicionais

Usando chaves e indentação para deixar mais claro:

```
if (cond1){  
    if (cond2)  
        comando1;  
    else  
        comando2;  
}
```


Comandos condicionais

```
int main(){
    int a = 5;

    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
            printf("b");
        else
            printf("c");
    }
}
```

O que será impresso?

Comandos condicionais

```
int main(){
    int a;
    a = -12;
    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
            printf("b");
        else
            printf("c");
    }
}
```

O que será impresso?

Comandos condicionais

```
int main(){
    int a;
    a = 9;
    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
            printf("b");
        else
            printf("c");
    }
}
```

O que será impresso?

Exercícios

A solução abaixo está correta para classificar um número como par e menor que 100, ou par e maior ou igual a 100, etc, como no exemplo visto anteriormente?

```
#include <stdio.h>

int main(){
    int a;
    printf("Digite um número inteiro:");
    scanf("%d", &a);
    if( ( a % 2 == 0) && (a<100) )
        printf("O número é par e menor que 100\n");
    else if( a>=100 )
        printf("O número é par e maior ou igual a 100\n");

    if( ( a % 2 != 0) && (a<100) )
        printf("O número é ímpar e menor que 100\n");
    else if( a>=100)
        printf("O número é ímpar e maior que 100\n");
}
```

Exercícios

- Escreva um programa que lê um número inteiro do teclado e imprime "SIM" se o número for par e maior do que 10, ou for ímpar e menor do que 50. Caso contrário o programa deve imprimir "NAO".

Exercícios

- Escreva um programa lê três números e imprime o maior deles.

Exercícios

- Escreva um programa lê três números e os imprime em ordem (ordem crescente).