

MC-102 — Aula 05

Comandos Condicionais

Instituto de Computação – Unicamp

Segundo Semestre de 2011

Roteiro

- 1 Expressões relacionais
- 2 Expressões lógicas
- 3 Comandos condicionais
- 4 O comando switch

Expressão

- Já vimos que constantes, variáveis e endereços de variáveis são expressões.

Exemplo

```
a = 10;  
a = b;  
endereco = &a;
```

- Vimos também que operações aritméticas também são expressões.

Exemplo

```
a = 2 + 2;  
a = 10 / (float) 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.

Expressões relacionais

Para não esquecer os valores possíveis de uma expressão relacional, use:

S1M NÃO

Expressões relacionais

Os operadores relacionais são:

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

Expressões relacionais

- $\langle \textit{expressao} \rangle == \langle \textit{expressao} \rangle$: Retorna verdadeiro quando as expressões forem iguais.
Ex: $a == b$
- $\langle \textit{expressao} \rangle != \langle \textit{expressao} \rangle$: Retorna verdadeiro quando as expressões forem diferentes.
Ex: $a != b$

Expressões relacionais

- $\langle \textit{expressao} \rangle > \langle \textit{expressao} \rangle$: Retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.
Ex: $a > b$
- $\langle \textit{expressao} \rangle < \langle \textit{expressao} \rangle$: Retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.
Ex: $a < b$

Expressões relacionais

- $\langle \textit{expressao} \rangle \geq \langle \textit{expressao} \rangle$: Retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.
Ex: $a \geq b$
- $\langle \textit{expressao} \rangle \leq \langle \textit{expressao} \rangle$: Retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.
Ex: $a \leq b$

Expressões lógicas

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

Expressões lógicas

- $\langle \text{expressao} \rangle \ \&\& \ \langle \text{expressao} \rangle$: Retorna verdadeiro quando ambas as expressões são verdadeiras. Sua tabela verdade é:

Op_1	Op_2	Ret
V	V	V
V	F	F
F	V	F
F	F	F

Exemplo

```
a == 0 && b == 0
```

Expressões lógicas

- $\langle \textit{expressao} \rangle \ || \ \langle \textit{expressao} \rangle$: Retorna verdadeiro quando pelo menos uma das expressões é verdadeiras. 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

Exemplo

```
a == 0 || b == 0
```

Expressões lógicas

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

<i>Op₁</i>	<i>Ret</i>
V	F
F	V

Exemplo

!(a == 0)

Simplificações úteis

- $!(a == b)$ é equivalente a $a != b$
- $!(a != b)$ é equivalente a $a == b$
- $!(a > b)$ é equivalente a $a <= b$
- $!(a < b)$ é equivalente a $a >= b$
- $!(a >= b)$ é equivalente a $a < b$
- $!(a <= b)$ é equivalente a $a > b$

Lei de De Morgan 1

- $!a \ \&\& \ !b$ é equivalente a $!(a \ || \ b)$

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

Lei de De Morgan 2

- $!a \ || \ !b$ é equivalente a $!(a \ \&\& \ b)$

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

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.



Comandos condicionais

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

```
if (expressão lógica)  
    comando; ou  
if (expressão lógica) {  
    comandos  
}
```

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

Bloco de comandos

- É um conjunto de instruções agrupadas.
- Limitada pelos caracteres { e }.
- Declaração de variáveis “locais”:
 - Devem ser sempre declaradas antes de qualquer outro comando.
 - São válidas somente dentro do bloco.

Exemplo

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

Comandos condicionais

O programa abaixo determina se um valor é ímpar.

```
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 lógica) {  
    comandos executados se a expressão é verdadeira  
} else {  
    comandos executados se a expressão é falsa  
}
```

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?

Comandos condicionais

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

Quando o comando2 é executado?

Decisão simples e decisão múltipla

- Dependendo do problema proposto, o programa pode ser formado por um conjunto muito grande de comandos `if` e expressões lógicas.

Exemplo

Faça um programa que, dado um RA, emite uma mensagem se o aluno estiver matriculado em uma turma de MC102.

Decisão simples

Para apenas um aluno, a solução seria:

```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129) {  
        printf("O aluno %d está matriculado\n", a);  
    }  
}
```

Decisão múltipla

Para dois alunos, a solução seria:

```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129 || a == 16267) {  
        printf("O aluno %d está matriculado\n", a);  
    }  
}
```

Decisão múltipla

- Problema: cada turma de MC102 possui cerca de 60 alunos e temos 14 turmas neste semestre.

```
if (a == 2582 || a == 10129 ||  
    a == 16267 || ...  
    a = 962185) {  
    printf("O aluno %d está matriculado\n", a);  
}
```

- Teríamos muitas condições a serem testadas.

Decisão simples e decisão múltipla

Exemplo 2

- Faça um programa que, dado um RA, mostre o nome desse aluno.

Decisão simples

Para apenas um aluno, a solução seria:

```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129) {  
        printf("Maria Cândida Moreira Telles\n");  
    }  
}
```

Decisão múltipla

```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129)  
        printf("Maria Cândida Moreira Telles\n");  
    if (a == 33860)  
        printf("Larissa Garcia Alfonsi\n");  
    if (a == 33967)  
        printf("Leonardo Kozlowiski Kenupp\n");  
}
```

Decisão múltipla

- Porém todos os testes condicionais serão executados!!
- Quando apenas uma de várias alternativas é verdadeira podemos usar a construção if-else-if:

```
if (<condição1>
    <comando>
else if (<condição2>)
    <comando>
...
else if (<condiçãoN>)
    <comando>
```



```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129)  
        printf("Maria Cândida Moreira Telles\n");  
    else if (a == 33860)  
        printf("Larissa Garcia Alfonsi\n");  
    else if (a == 33967)  
        printf("Leonardo Kozlowiski Kenupp\n");  
}
```

Decisão múltipla

- Na construção if-else-if quando uma condição é verdadeira, o bloco de comandos correspondente será executado.
- Após a execução do bloco de comandos as outras alternativas não serão testadas.
- O ultimo else (sem if) pode ser utilizado como uma opção padrão quando nenhuma das condições dos ifs é verdadeira.

```
main () {  
    int a;  
    scanf("%d", &a);  
    if (a == 10129) {  
        printf("Maria Cândida Moreira Telles\n");  
    }else if (a == 33860){  
        printf("Larissa Garcia Alfonsi\n");  
    }else if (a == 33967){  
        printf("Leonardo Kozlowiski Kenupp\n");  
    }else  
        printf("aluno não cadastrado\n");  
}
```

O comando `switch`

- O objetivo do comando `switch` é simplificar uma expressão onde uma variável **inteira** ou **caracter** deve fazer diferentes operações dependendo exclusivamente de seu valor.

Sintaxe

```
switch (variável inteira) {  
    case valor: comandos  
    break;  
    case valor: comandos  
    break;  
}
```

O comando switch

```
switch(a) {  
case 10129:  
    printf("Maria Cândida Moreira Telles\n");  
    break;  
case 33860:  
    printf("Larissa Garcia Alfonsi\n");  
    break;  
case 33967:  
    printf("Leonardo Kozlowiski Kenupp\n");  
    break;  
}
```

O comando `switch`

- Os comandos começam a ser executados a partir do ponto onde o valor da variável corresponde ao valor antes dos dois pontos (:).
- Executa todos os comandos até que encontre um comando `break` ou que chegue ao final do bloco de comandos do `switch`

Valor padrão

- Você pode utilizar, ao invés de um valor, o valor `default`. A execução dos comandos inicia no comando `default` se nenhum outro valor for correspondente ao valor da variável.

Sintaxe

```
switch (variável inteira) {  
    valor: comandos break;  
    default: comandos  
}
```

Valor padrão

```
switch(a) {  
case 10129:  
    printf("Maria Cândida Moreira Telles\n");  
    break;  
case 33860:  
    printf("Larissa Garcia Alfonsi\n");  
    break;  
default:  
    printf("O aluno não está matriculado\n");  
}
```


Exercícios

- Escreva um programa que ordene três números. Tente encontrar uma versão com apenas 3 comandos `if`.
- Escreva um programa que, dado duas datas, determine qual delas ocorreu cronologicamente antes em relação a outra. Cada data é composta de 3 números inteiros, um representando o ano, outro o mês e outro o dia.
- Dada uma letra, escreva na tela se essa letra é ou não uma vogal (pode considerar apenas letras minúsculas).

Exercícios

- Escreva um programa que mostre na tela um *menu* de pratos (pelo menos 5), cada um associado a um número.

Prato 1 - Miojo

Prato 2 - Ensopado

- Quando um número é selecionado, o programa deve exibir uma breve descrição do prato. Por exemplo, ao digitar 1, o programa mostra: “Macarrão instantâneo”