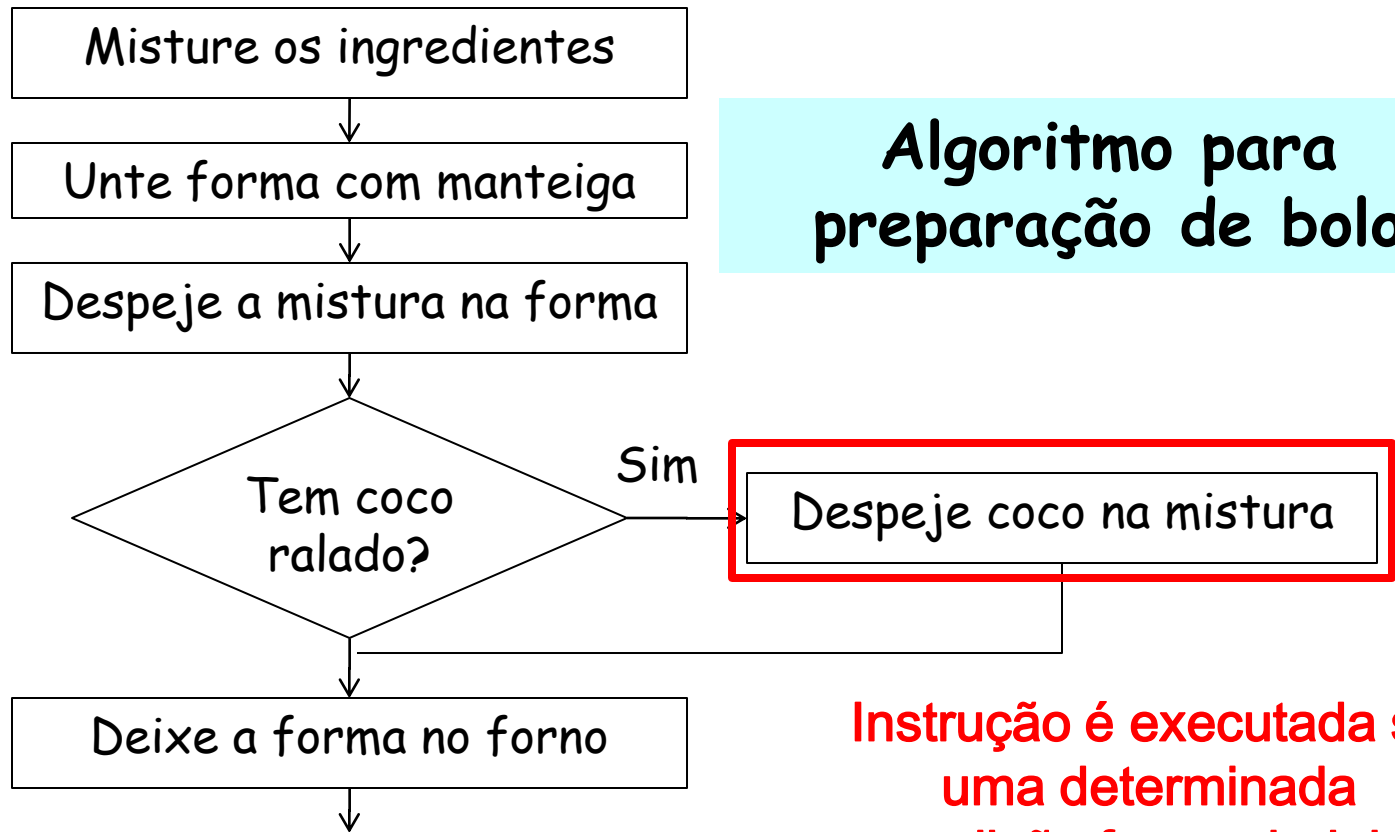


Introdução à Programação



Expressões Booleanas e Comandos Condicionais

Comandos Condicionais



Instrução é executada se
uma determinada
condição for verdadeira

Comandos Condicionais

Tópicos da Aula

- ◆ Hoje vamos acrescentar comportamentos mais complexos a programas
 - Tomada de decisão e Mudança de fluxo de controle
 - Expressões booleanas (lógicas)
 - Comandos Condicionais
 - *if-else*
 - *switch*
 - Operador condicional

Instruções de Controle

- ◆ Toda linguagem de programação precisa oferecer pelo menos três formas básicas de controle:
 - executar uma sequência de instruções
 - realizar testes para decidir entre ações alternativas
 - repetir uma sequência de instruções

Hoje, aprenderemos a realizar testes para decidir quais instruções executar

Modificando Fluxo de Controle

- ◆ A ordem de execução de um programa é denominado **fluxo de controle**
- ◆ Exceto quando especificado de outra forma, a ordem de execução é linear, isto é uma instrução após a outra em seqüência
- ◆ Alguns comandos em programação nos permitem:
 - Decidir se a execução de uma instrução deve ou não ser feita
- ◆ Útil para que certas instruções só sejam executadas sob determinadas condições
- ◆ Esta tomada de decisão é baseada em **expressões booleanas**

Expressão Booleana

- ◆ O resultado da avaliação de uma **expressão booleana** é **ou verdadeiro ou falso**
 - Em C, **NÃO** existe o tipo de dado boolean
 - Verdadeiro é representado como 1
 - Falso é representado como 0
- ◆ Uma expressão booleana é composta de operandos booleanos (lógicos) e operadores **relacionais e/ou lógicos**

Operadores Relacionais

◆ São usados para fazer comparações

Operador	Ação
<	<i>menor que</i>
>	<i>maior que</i>
<=	<i>menor ou igual que</i>
>=	<i>maior ou igual que</i>
==	<i>igual a</i>
!=	<i>diferente de</i>

Resultado de Comparação

Falso ou Verdadeiro

4 < 5 é verdadeiro (valor 1)

3 >= 10 é falso (valor 0)

Operadores Lógicos (Booleanos)

- ◆ São usados para combinar comparações
 - Operam sobre valores booleanos (0 ou 1)

Operador	Ação
&&	<i>E</i>
	<i>Ou</i>
!	<i>Negação</i>

Resultado da Avaliação

```
int  a , b ;  
int  c = 23 ;  
int  d = 27 ;  
a = ( c < 20 ) || ( d > c ) ;  
b = ( c < 20 ) && ( d < c ) ;
```

Qual será o valor
de a e b?

a = 1

b = 0

Expressões Booleanas

◆ Operadores lógicos e expressões:

● **x && y**

● **x || y**

● **!x**

◆ Operadores && e || são ditos *short-circuited*

- Operandos da direita só são avaliados, se necessário
- Existem os operadores bit a bit **&** e **|**

x e **y** são
expressões
booleanas

As expressões resultantes
são **booleanas**, gerando
verdadeiro (1) ou **falso (0)**
como resultado
da avaliação

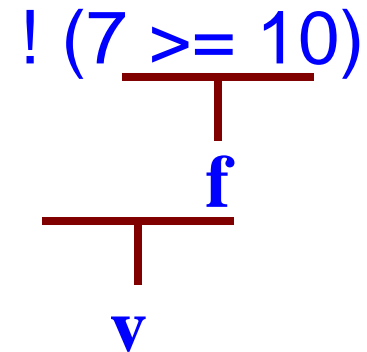
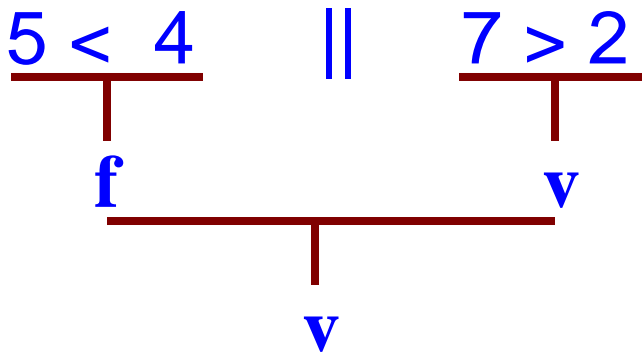
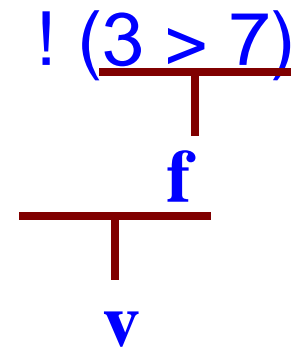
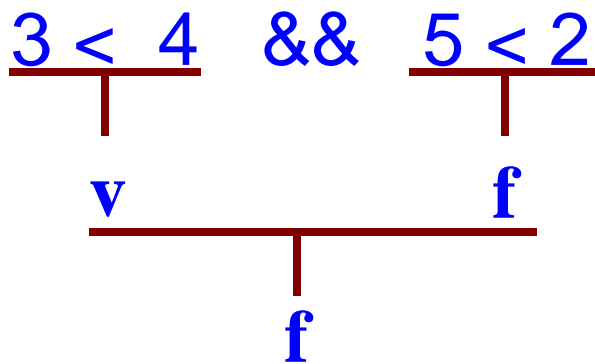
Tabela Verdade

- ◆ Uma tabela verdade contém todas as combinações verdadeiro-falso de uma expressão booleana

a	b	a && b	a b	!a
verdadeiro	verdadeiro	verdadeiro	verdadeiro	falso
verdadeiro	falso	falso	verdadeiro	falso
falso	verdadeiro	falso	verdadeiro	verdadeiro
falso	falso	falso	falso	verdadeiro

Expressões Booleanas

◆ Exemplos



Expressões Booleanas

```
{    ...  
    int b, c;  
    b = 1;  
    c = !b;  
    c = !(1 || b) && c;  
    b = c || !(b);  
    ...  
}
```

Parênteses são
usados para
evitar
ambigüidades

Qual o valor
de b neste
ponto?

1 (verdadeiro)

Expressões Booleanas

```
int b, c;
```

```
b = 1 || 0;
```

```
c = 1 && b;
```

```
b = b == c;
```

Qual o valor
de b aqui?

1 (verdadeiro)

Comandos Condicionais

- ◆ Um **comando condicional** nos permite escolher qual deve ser a próxima instrução executada em um programa
- ◆ A execução de uma determinada instrução depende de uma condição (expressão booleana)
- ◆ A linguagem C oferece 3 tipos de comandos condicionais:
 - **if – else**
 - **switch**
 - **comando ternário**

O comando if-else

```
if (expressaoBooleana) {  
    comandos  
} else {  
    outros comandos  
}
```

Se a avaliação de **expressaoBooleana** retornar **verdadeiro**, **comandos** são executados, caso contrário, executa-se **outros comandos**

Exemplo do if-else

```
#include <stdio.h>

int main ( )
{ float  n1, n2, n3, m;
  printf ("\nEntre com 3 notas " ) ;
  scanf ("%f %f %f", &n1, &n2, &n3);
  m  =  (n1 + n2 + n3 ) / 3 ;
  if (m >= 7.0)    {
    printf ("\n Aluno aprovado. ") ;
    printf (" Média igual a %f " , m) ;
  }else {
    printf ("\n Aluno reprovado. ");
    printf (" Média igual a %f " , m) ;
  }
  return 0;
}
```


Variações do comando if-else

```
if (expressaoBooleana) {  
    comandos  
}
```

```
if (expressaoBooleana)  
    comando;
```

```
if (expressaoBooleana)  
    comando;  
else outroComando;
```

Se a avaliação da expressão retornar **falso**, não executa-se nada

O uso do bloco só é necessário caso queira-se executar mais de um comando

Exemplo – if sem else

```
#include <stdio.h>

int main ( )
{ int resposta ;
  printf ("\n Qual o valor de 10 + 14? ");
  scanf ("%d", &resposta);
  if (resposta == 10 + 14)
    printf ("\n Resposta correta ! ");
  return 0;
}
```

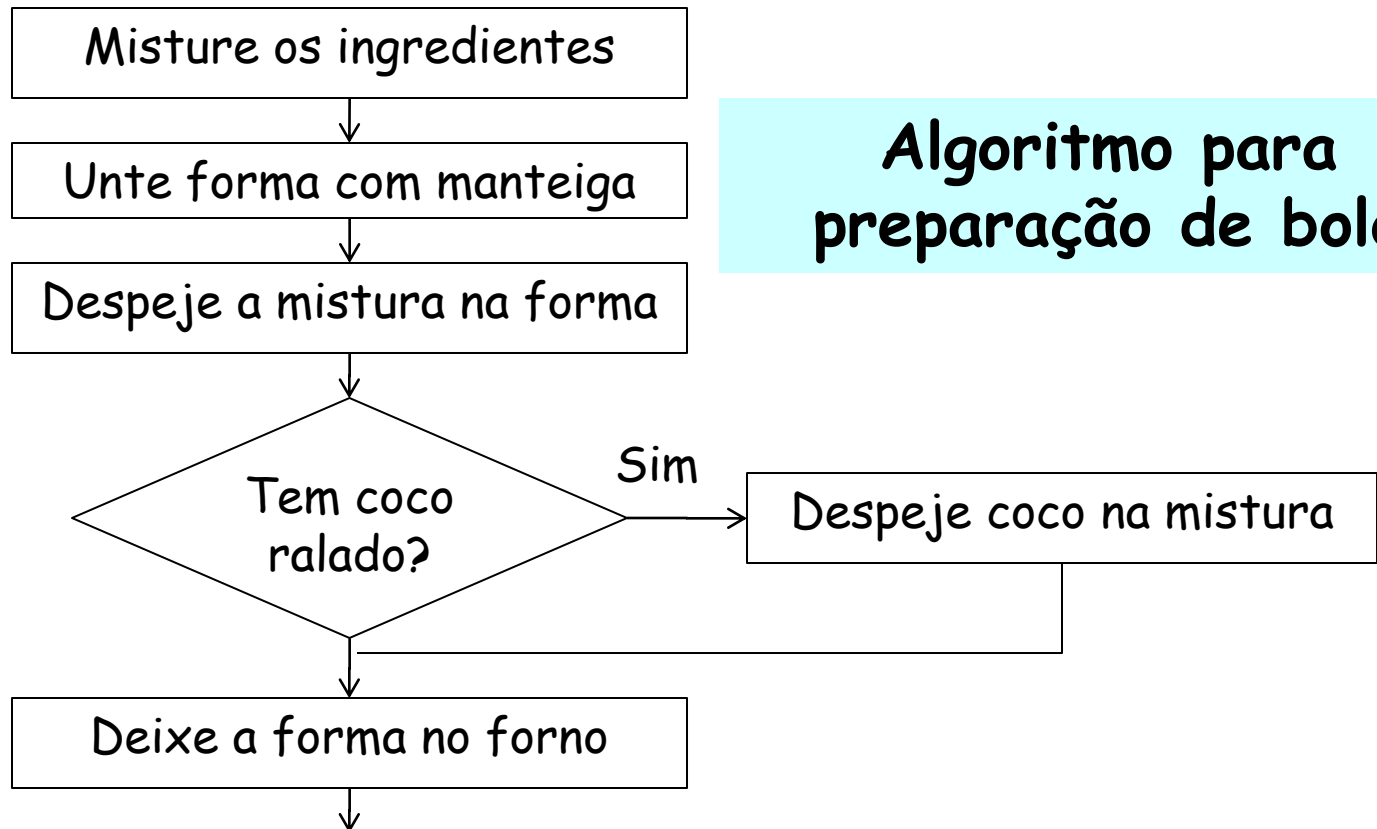
else não é obrigatório

Exemplo – if-else com único comando

```
#include <stdio.h>

int main ( )
{ int num ;
  printf ("\nDigite um número: ");
  scanf ("%d", &num);
  if (num < 0)
    printf ("\n Número é negativo ! ");
  else
    printf ("\n Número é positivo ! ");
  return 0;
}
```

Problema – Imprimir Receita de Bolo



- ◆ Escreva um programa em C que leia do usuário um caractere, que informa se tem coco ralado ('S' ou 'N'), e depois imprime os passos para fazer um bolo.

Aninhando if-else

```
if (expressaoBooleana)  
  if (expressaoBooleana)  
    comando;  
  else outroComando;
```

if-else aninhado

O comando dentro do if ou else
pode ser outro if

Num aninhamento, o **else** é associado ao mais recente **if** sem else

Exemplo if-else Aninhado

- ◆ Exercício: O que está errado neste programa?

```
#include <stdio.h >
int main ()
{ int temp ;
  printf ("\n Digite a temperatura: " ) ;
  scanf ("%d", &temp) ;
  if (temp < 30)
    if (temp > 20)
      printf ("\n Temperatura agradável " ) ;
  else printf ("\n Temperatura muito quente " ) ;
  return 0 ;
}
```

Exemplo if-else Aninhado

◆ Exercício: Corrigindo o programa

```
#include <stdio.h >
int main ()
{ int temp ;
  printf ("\n Digite a temperatura: " ) ;
  scanf ("%d", &temp) ;
  if (temp < 30) {
    if (temp > 20)
      printf ("\n Temperatura agradável " ) ;
  } else printf ("\n Temperatura muito quente " ) ;
  return 0 ;
}
```

else associado ao if certo

Encadeando comandos if-else

```
if (expressaoBooleana) {  
    comandos  
} else if (expressaoBooleana') {  
    comandos'  
} else {  
    comandos' '  
}
```


Tomando Múltiplas Decisões

- ◆ O comando `if-else` é útil para a escolha de uma entre duas alternativas
- ◆ Quando mais de duas alternativas são necessárias, pode ficar deselegante utilizar vários `if-else` encadeados
 - Para estes casos o comando `switch` pode ser a melhor opção

O Comando switch

```
switch (expressao) {  
    case rotulo1:  
        Comandos1  
        break;  
    case rotulo2:  
        Comandos2  
        break;  
    ...  
    default:  
        Comandos  
}
```

Para executar um **switch**

- Avalia-se **expressao**
- Executa-se os comandos do **case** cujo rótulo é igual ao valor resultante da expressão
- Executa-se os comandos de **default** caso o valor resultante não seja igual a nenhum rótulo

Restrições do Comando switch

```
switch (expressao) {  
    case rotulo1:  
        Comandos1  
        break;  
    case rotulo2:  
        Comandos2  
        break;  
    ...  
    default:  
        Comandos  
}
```

- ◆ O tipo de **expressao** só pode ser :
 - **Inteiro ou caractere**
- ◆ Os rótulos são constantes diferentes
- ◆ Existe no máximo uma cláusula **default** (é opcional)
- ◆ Os tipos dos rótulos têm que ser o mesmo de **expressao**

Variações do Comando `switch`

```
switch (expressao) {  
    case rotulo1:  
        Comandos1  
        break;  
    case rotulo2:  
        Comandos2  
        break;  
    ...  
    default:  
        Comandos  
}
```

- ◆ Vários rótulos podem estar associados ao mesmo comando
- ◆ Os comandos **break** são opcionais:
 - Sem o **break** a execução dos comandos de um rótulo continua nos comandos do próximo, até chegar ao final ou a um **break**

Exemplo de switch

- ◆ Calcular a diferença, o produto, o quociente ou a soma de dois números, dependendo da operação escolhida e imprimir o resultado.

```
# include      "stdio.h"
int main ()
{
char operador;
float a, b ;
float result = 0.0;
printf ("\n Informe os 2 números e a operação ");
scanf ("%f %f %c", &a, &b, &operador);
```

Exemplo de switch (cont..)

```
switch (operador) {  
    case '+':    result = a + b;  
                break;  
    case '-':    result = a - b;  
                break;  
    case '*':    result = a * b;  
                break;  
    case '/':    result = a / b;  
                break;  
    default :    printf("\nOperador invalido");  
}  
printf ("\nResultado igual a %f ", result);  
return 0 ;  
}
```

Operador Condicional - ?

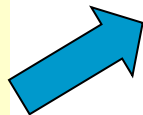
◆ Forma Geral do ?

condição ? expressão1 : expressão2

◆ Substitui construções do tipo:

```
if ( a > b ) {  
    maximo = a;  
}  
else {  
    maximo = b;  
}
```

maximo = a > b ? a : b;



Comando Ternário de Decisão - ?

♦ Exercício: Considere as seguintes declarações

```
char a = 'a' , b = 'b' ; /* b tem valor 98 */  
int i = 1, j = 2 ;  
double x = 7.07 ;
```

Expressão	Valor
$i == j ? a - 1 : b + 1$	'c'
$j \% 3 == 0 ? i + 4 : x$	7.07
$j > 1 ? j - 1 : j + 1$	1
$j \% 3 != 0 ? i + 4 : x$	5

Resumindo ...

- ◆ Necessidade de Mecanismos de Tomadas de Decisão em um Programa
- ◆ Mudança de Fluxo de Controle
- ◆ Expressões Booleanas
- ◆ Comandos Condicionais em C
 - if-else
 - switch
 - Operador Condicional