

Fluxogramas e Condicionais

Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT

Agenda

- 1 Objetivos
- 2 Estrutura de um Programa
- 3 Fluxogramas
- 4 Biblioteca `<math.h>`
- 5 Operadores Relacionais
- 6 Operadores Lógicos
- 7 Comandos de Decisão
- 8 Exercícios

Objetivos

- Apresentar o uso de fluxogramas.
- Demonstrar o uso da biblioteca `math.h`.
- Explicar o uso dos operadores relacionais, lógicos e de decisão.

Estrutura de um Programa em C

A estrutura básica de qualquer programa em C é a seguinte.

```
<inclusão_de_bibliotecas>
```

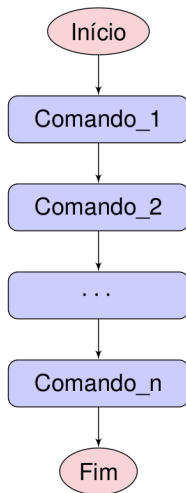
```
int main(<declaração_dos_parâmetros>) {  
    instrução_1;  
    instrução_2;  
    instrução_3;  
    ...  
    instrução_n;  
    return 0;  
}
```

Fluxogramas

- Os fluxogramas são representações gráficas dos programas.
- São utilizados para nos ajudar a compreender um programa.
- Não estão associados a uma linguagem específica.
- Apresentam a lógica do algoritmo e não as instruções da linguagem.
- Utilizam diferentes tipos de blocos para indicar os comandos (entradas, saídas, processamentos, decisões, etc) e setas para indicar a sequência de execução.

Fluxogramas

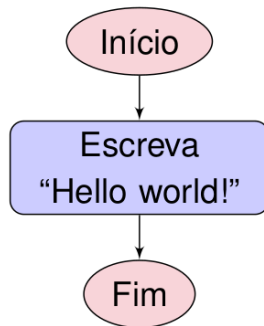
Fluxograma de um programa em C/C++:



Fluxogramas

Programa Hello World:

```
// Programa Hello World
#include<stdio.h>
int main() {
    // comentario explicativo
    printf("Hello World");
    return 0;
}
```



Fluxogramas

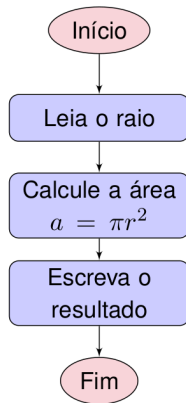
Exemplo 1

Faça um programa em C para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

Fluxogramas

Exemplo 1

Fluxograma da solução do Exemplo 1:



Fluxogramas

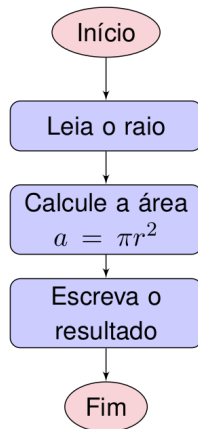
Exemplo 1

Programa Área Círculo:

```
// Programa que calcula a area de um circulo
#include<stdio.h>
int main() {
    // declaracao da constante Pi
    const double PI = 3.141592;
    double raio;

    printf("Digite o raio do circulo: ");
    scanf("%lf", &raio);

    // calculando e imprimindo a area
    double area = PI * raio * raio;
    printf("\nArea do circulo: %lf\n", area);
    return 0;
}
```



O Qualificador const

- A palavra-chave `const` assegura que a variável associada não será alterada em todo o programa.
- Esse qualificador é indicado para declarar valores **constantes**.
- *Obrigatoriamente*, as variáveis associadas ao qualificador `const` devem ser inicializadas.

Dúvida

- Não existe área negativa.
- Portanto, o programa não pode calcular a área se o valor do raio for negativo.
- Como saber se o valor do raio digitado é positivo?
 - ▶ Responderemos no final da aula...

Biblioteca <math.h>

- Como calcular πr^2 ?
 - ▶ `double area = PI * raio * raio;`
- A linguagem C não possui um operador para potência, mas possui uma biblioteca com diversas funções matemáticas, para usá-la devemos incluir a biblioteca `math.h`
- A função para potência é a `pow()`, sintaxe:
 - ▶ `double pow(double base, double expoente);`
- Exemplo

```
// Programa que calcula a area de um circulo
#include<math.h>

...
double area = PI * pow(raio, 2);
...
```

Biblioteca `<math.h>`

Parte I

Algumas funções matemáticas disponíveis na biblioteca `#include <math.h>`. Para usá-las é necessário: `#include <math.h>`

Função	Descrição	Exemplo
<code>double ceil(x)</code>	arredonda x para cima	<code>ceil(9.1) → 10.0</code>
<code>double floor(x)</code>	arredonda x para baixo	<code>floor(9.8) → 9.0</code>
<code>double round(x)</code>	arredonda x	<code>round(9.5) → 10.0</code> <code>round(9.4) → 9.0</code>
<code>double trunc(x)</code>	retorna a parte inteira de x	<code>trunc(9.8) → 9.0</code>

Biblioteca <math.h>

Parte I

Exemplo: Dada a tabela abaixo com os os valores de x , escreva os valores retornados pelas funções.

x	<code>round(x)</code>	<code>floor(x)</code>	<code>ceil(x)</code>	<code>trunc(x)</code>
2.3	2.0	2.0	3.0	2.0
3.8	4.0	3.0	4.0	3.0
5.5	6.0	5.0	6.0	5.0
-2.3	-2.0	-3.0	-2.0	-2.0
-3.8	-4.0	-4.0	-3.0	-3.0
-5.5	-6.0	-6.0	-5.0	-5.0

Funções para Potências:

Função	Descrição	Exemplo
<code>double pow(x, y)</code>	x elevado a y: x^y	<code>pow(3, 2) → 9.0</code>
<code>double sqrt(x)</code>	raiz quadrada de x: \sqrt{x}	<code>sqrt(25) → 5.0</code>
<code>double cbrt(x)</code>	raiz cúbica de x: $\sqrt[3]{x}$	<code>cbrt(27) → 3.0</code>

Funções trigonométricas:

Função	Descrição	Exemplo
double <code>cos(x)</code> *	retorna o cosseno x	<code>cos(1.047)</code> → 0.5
double <code>sin(x)</code> *	retorna o seno x	<code>sin(1.571)</code> → 1.0
double <code>tan(x)</code> *	retorna a tangente x	<code>tan(0.785)</code> → 1.0
double <code>acos(x)</code> **	retorna o arco cosseno	<code>acos(0.5)</code> → 1.047
double <code>asin(x)</code> **	retorna o arco seno	<code>asin(1.0)</code> → 1.571
double <code>atan(x)</code> **	retorna o arco tangente	<code>atan(1.0)</code> → 0.785

*: valores em radianos

**: valores de x entre $[-1, 1]$

Funções Exponencias e Logarítmicas:

Função	Descrição	Exemplo
<code>double exp(x)</code>	retorna exponencial de x: e^x	<code>exp(5)</code> → 148.4
<code>double log(x)</code>	logaritmo natural de x: $\ln(x)$	<code>log(5.5)</code> → 1.7
<code>double log10(x)</code>	logaritmo de x: $\log(x)$	<code>log10(1000)</code> → 3.0

Operadores Relacionais

- Permite a um programa realizar uma ação alternativa, a partir de um resultado **verdadeiro** ou **falso** produzido por uma condição.
- As **condições** são formadas utilizando-se os operadores de igualdade e os operadores relacionais.
- Ambos operadores de igualdade têm o mesmo nível de precedência, o qual é inferior ao dos operadores relacionais, e associam-se da esquerda para a direita.

Operadores Relacionais


Operadores de igualdade e relacionais:

Operador algébrico de igualdade ou relacional padrão	Operador de igualdade ou relacional em C++	Exemplo de condição em C++	Significado da condição em C++
<i>Operadores relacionais</i>			
>	>	<code>x > y</code>	x é maior que y
<	<	<code>x < y</code>	x é menor que y
≥	>=	<code>x >= y</code>	x é maior que ou igual a y
≤	<=	<code>x <= y</code>	x é menor que ou igual a y
<i>Operadores de igualdade</i>			
=	==	<code>x == y</code>	x é igual a y
≠	!=	<code>x != y</code>	x não é igual a y

Operadores Relacionais

Erros Comuns

Confundir o operador de igualdade `==` com o operador de atribuição `=`.

`total = x + 10;`

avalia a expressão (direita)
e atribui o resultado
à variável (esquerda)

`total == x + 10;`
verifica se a expressão da
direita é IGUAL a expressão
da esquerda (vice-versa).

Operadores Lógicos

- No século 18, George Boole, matemático e filósofo britânico, desenvolveu um sistema algébrico lógico, que passou a ser conhecido como Álgebra de Boole.
- Base para a lógica dos computadores digitais modernos.
- Expressões lógicas (expressões booleanas) possuem o valor **true** ou **false**.
- Em C os inteiros também são usados como valores booleanos:
 - ▶ qualquer valor não nulo (**1**) representa **true** e **0** representa **false**.

Operadores Lógicos

Expressões booleanas

- na matemática, podemos restringir uma temperatura a um intervalo fechado,
 $0 \leq temp \leq 100$
- em C não podemos representar essa expressão por $0 \leq temp \leq 100$.
- A expressão $0 \leq temp \leq 100$ é uma expressão válida na linguagem C.
 - ▶ no entanto, o resultado não será o esperado.
- por exemplo, suponha que `temp = 150`;
 - ▶ não está no intervalo definido, logo esperamos que o resultado da expressão seja falso.

Operadores Lógicos

Expressões booleanas

- Os operadores relacionais são associativos a esquerda. A expressão será processada da seguinte forma:

$$\underbrace{0 \leq 150}_{(a)} \leq 100$$

- A expressão (a) resulta em **true**, que é representado em C pelo inteiro 1 (ou outro inteiro diferente de 0).
- Assim, na segunda etapa da avaliação, a expressão resulta em:

$$1 \leq 100$$

- que também é verdadeira e resulta em **true**.
- Entretanto, a expressão original deveria resultar **false**, como na matemática.

Operadores Lógicos

Expressões booleanas

- Para solucionar esse problema, reescrevemos a desigualdade como:

$$(temp \geq 0) \&\& (temp \leq 100)$$

- onde $\&\&$ é um operador lógico.
- Utilizamos os operadores lógicos para combinar expressões booleanas formando, assim, expressões booleanas compostas

Operadores Lógicos

Expressões booleanas

Operador	Expressão	Nome	Descrição
!	!p	NÃO (negação)	!p é falso, se p é verd.; !p é verd., se p é falso.
&&	p && q	E (conjunção)	p && q é verdadeiro, se ambos, p e q são verd.; e falso, caso contrário.
	p q	OU (disjunção)	p q é verdadeiro, se p , q ou ambos é verd.; e falso, caso contrário.

Operadores Lógicos

Tabela Verdade

p	$\neg p$
true	false
false	true

p	q	$p \ \&\& \ q$	$p \ \ q$
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Operadores Lógicos

Precedência

Operator	Priority	Associativity
!, ~	highest	Right
/, *, %		Left
+, -		Left
<, >, <=, >=		Left
==, !=		Left
&		Left
^		Left
		Left
&&		Left
		Left
=, +=, *=, ...	lowest	Right

Comandos de Decisão

- Permite a um programa realizar uma ação alternativa, a partir de um resultado **verdadeiro** ou **falso** produzido por uma condição.
- As condições são formadas utilizando-se os operadores de igualdade e os operadores relacionais.
- Ambos operadores de igualdade têm o mesmo nível de precedência, o qual é inferior ao dos operadores relacionais, e associam-se da esquerda para a direita.

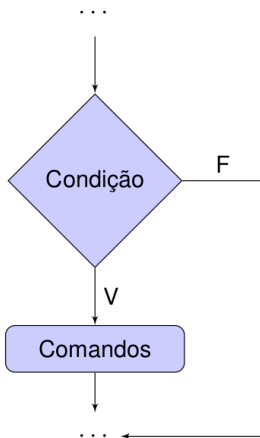
Comandos de Decisão

Comando **if**:

- consiste de uma palavra-chave **if** seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for **verdadeira**.
- O corpo de um comando **if** pode conter uma única instrução terminada por **ponto-e-vírgula** ou várias instruções entre **chaves**.

Comandos de Decisão

Fluxograma



Comandos de Decisão

Sintaxe

```
1  if ( <expressão_de_teste> )  
2      instrução_única;
```

OU

```
1  if ( <expressão_de_teste> )  
2  {  
3      instrução1;  
4      instrução2;  
5      instrução3;  
6      ...  
7  }
```


Comandos de Decisão

Exemplo 1

Faça um programa na linguagem C para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário. Altere o programa anterior para calcular a área somente se o valor do raio for positivo.

Fluxogramas

Exemplo 2

```
// Programa que calcula a area de um circulo
#include<stdio.h>
int main() {
    // declaracao da constante Pi
    const double PI = 3.141592;
    double raio;

    printf("Digite o raio do circulo: ");
    scanf("%lf", &raio);

    // calculando e imprimindo a area
    if (raio >= 0) {
        double area = PI * raio * raio;
        printf("\nArea do circulo: %lf\n", area);
    }
    return 0;
}
```

Exercícios

Maior Número

Faça um programa na linguagem C que leia dois números inteiros e verifique qual deles é maior. Para isso, imprima uma mensagem informando qual deles é o maior. Exemplo de saída caso o usuário digite 10 e 20:

20 é o maior número

Fim

Fim

Referências

- Slides prof. Túlio Toffolo
- http://www3.decom.ufop.br/toffolo/pt-br/ensino/bcc201_20201/

Fluxogramas e Condicionais

Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT