

Linguagem C

Estrutura Condicional

Prof. Edwar Saliba Júnior
Fevereiro de 2011



Falso e Verdadeiro em C

- Falso: valor igual a 0 (zero);
- Verdadeiro: valor diferente de zero: 2, -83, 0.1, -7.329, 529;
- Resultado de comparações numéricas em C:
 - Se for igual a zero, então FALSO;
 - Se for diferente de zero, então VERDADEIRO.

Operadores Lógicos e Relacionais

Operador

Significado

>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
==	igual a
!=	diferente de
&&	condição "E"
	condição "OU"
!	negação

Prioridade dos Operadores

	Operador
Maior	() [] ->
	! ~ ++ --(tipo) * & sizeof
	* / %
	+ -
	<< >>
	<<= >>=
	== !=
	&
	^
	&&
	?
	= += -= *= /=
Menor	,

Exercício

```
if (10 > 4 && ! (10 < 9) || 3 <= 4)
```

Qual o resultado da sentença acima?

Estrutura Condicional Simples

- Comando `if` (se), sintaxe:

```
if (expressão) {  
    comandos  
}
```

- Onde `expressão` deve gerar um resultado booleano, e quando este resultado é verdadeiro, os `comandos` entre `{ }` são executados;
- Se a `expressão` é falsa, então os `comandos` não serão executados. E o programa continua a sua execução normal, após a `}`. Ou seja, somente o teste do comando `if` é feito e nada mais.

Exemplo Estrutura Condicional Simples

```
main.c [x]
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float nota = 0;
7
8      printf("Digite a nota do aluno: ");
9      scanf("%f",&nota);
10
11     if(nota >= 60){
12         printf("\n\nAluno aprovado!");
13     }
14
15     return 0;
16 }
17
```

Estrutura Condicional Completa

- Comando `if / else` (se / senão), sintaxe:

```
if (expressão) {  
    comandos A  
}  
else {  
    comandos B  
}
```

- Onde `expressão` deve gerar um resultado booleano, e quando este resultado é verdadeiro, os comandos A entre { } são executados;
- Se a `expressão` gerar um resultado falso, então os comandos B, que estão entre { } do comando `else`, serão executados.

Exemplo Estrutura Condicional Completa

```
main.c [x]
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float nota = 0;
7
8      printf("Digite a nota do aluno: ");
9      scanf("%f",&nota);
10
11     if(nota >= 60){
12         printf("\n\nAluno aprovado!");
13     }
14     else{
15         printf("\n\nAluno reprovado!");
16     }
17
18     return 0;
19 }
20
```

Aninhamento `if / else`

```
*main.c
1  #include <stdio.h>
2
3  int main(){
4      float nota = 0;
5
6      printf("Digite a nota do aluno: ");
7      scanf("%f",&nota);
8
9      if(nota >= 60){
10         printf("\n\nAluno aprovado!");
11
12         if(nota >= 90){
13             printf("\nConceito A.");
14         }
15         else{
16             if(nota >= 80){
17                 printf("\nConceito B.");
18             }
19             else{
20                 if(nota >= 70){
21                     printf("\nConceito C.");
22                 }
23                 else{
24                     printf("\nConceito D.");
25                 }
26             }
27         }
28     }
29     else{
30         printf("\n\nAluno reprovado!");
31     }
32 }
```

- Dado o código-fonte ao lado, defina matematicamente os intervalos em que cada tipo de conceito é aplicado aos alunos.

Operador Ternário

- Sintaxe:

`(condição ? bloco_verdadeiro : bloco_falso);`

- O operador `?` : é uma maneira simplificada de escrever um `if...else`;
- Apesar de possuir a mesma funcionalidade, não se deve usar este operador quando os comandos envolvidos são complexos;
- Primeiramente a condição é avaliada. Dependendo do resultado o bloco respectivo será executado.

Exemplo de Uso do Operador Ternário

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float nota = 0;
7
8      printf("Entre com a nota final do aluno: ");
9      scanf("%f",&nota);
10
11     printf(nota >= 60 ? "Aluno aprovado!\n" : "Aluno reprovado!\n");
12
13     return 0;
14 }
15
```



Exercícios

- Faça um programa para ler dois valores do inteiros do teclado. Efetue a subtração do maior pelo menor número digitado, e apresente o resultado na tela.
- Faça um programa para ler três valores inteiros do teclado. Mostre na tela, o maior e o menor valor que foi digitado pelo usuário do *software*.

Bibliografia

- LAUREANO, Marcos. **Programação em C para ambiente Linux**. Disponível em: <<http://br-c.org/doku.php>>. Acesso em: 06 fev. 2011.
- MURTA, Cristina Duarte. *Slides da disciplina de Programação de Computadores I*. CEFET-MG, 2010.
- SENNE, Edson Luiz França. **Primeiro Curso de Programação em C**. 2. ed. Florianópolis: Visual Books, 2006.