

Aula 3 – Comando de Controle Condicional - IF

Operadores de Atribuição, Lógicos e Condicionais

Operadores

Operadores de Atribuição

- Uma das operações mais utilizadas em programação é a operação de atribuição “=”. Ela é responsável por armazenar determinado valor em uma variável. Em linguagem C, o uso do operador de atribuição “=” segue esta forma geral: ***nome_da_variável = expressão;***
 - float y = 3;
 - x = 5;

Operadores de Aritméticos

- Os operadores aritméticos são aqueles que operam sobre números (valores, variáveis, constantes ou chamadas de funções) e/ou expressões e têm como resultados valores numéricos. A **linguagem C** possui um total de 5 operadores aritméticos.

Operador	Significado	Exemplo
+	adição de dois valores	$z = x + y$
-	subtração de dois valores	$z = x - y$
*	multiplicação de dois valores	$z = x * y$
/	quociente de dois valores	$z = x / y$
%	resto de uma divisão	$z = x \% y$



O operador de resto da divisão (%) só é válido para valores inteiros (tipo **int** e **char**).

Vamos fazer um programa que solicite 2 números e realize essas operações?

Operadores de Relacionais

- Os operadores relacionais são aqueles que **operam sobre dois valores** (valores, variáveis, constantes ou chamadas de funções) e/ou expressões e verificam a magnitude (qual é maior ou menor) e/ou igualdade entre eles. A Linguagem C possui 6 operadores.

Operador	Significado	Exemplo
>	Maior do que	$x > 5$
>=	Maior ou igual a	$x \geq 10$
<	Menor do que	$x < 5$
<=	Menor ou igual a	$x \leq 10$
==	Igual a	$x == 0$
!=	Diferente de	$x != 0$



Os operadores relacionais são operadores de comparação de valores.

Operadores Lógicos

- Os operadores lógicos permitem representar situações lógicas unindo duas ou mais expressões relacionais simples em uma composta.

Operador	Significado	Exemplo
&&	Operador E	(x >= 0 && x <= 9)
	Operador OU	(a == 'F' b != 32)
!	Operador NEGAÇÃO	!(x == 10)

Tomando decisão: IF... Else

- Em linguagem C, o **comando IF** é utilizado quando for necessário escolher entre dois caminhos, ou quando se deseja executar um comando sujeito ao resultado de um teste.
- Se a expressão for verdadeira, será executada!
- Sintaxe:



Diferentemente da maioria dos comandos, não se usa o ponto e vírgula (;) depois da condição do comando **if**.

```
If (condição) {  
    Sequencia de comandos;  
}
```

```
printf ("Digite um numero: ");  
scanf("%d",&num);  
if(num > 10)  
    printf ("O numero e maior do que 10\n");
```


- O comando **else** pode ser entendido como um complemento do comando if. Ele auxilia o comando if na tarefa de escolher entre os vários caminhos a serem seguidos dentro do programa.
- Sintaxe:

```
If (condição) {  
    Primeira sequência de comandos;  
}  
else {  
    Segunda sequência de comandos;  
}
```



O comando **else** é opcional, e sua sequência de comandos somente será executada se o valor da condição que está sendo testada pelo comando **if** for **FALSA**.

Extra

Comando "puts"

- Exibe uma sequencia de caracteres (strings) no terminal. Automaticamente faz a quebra de linha (tecla escape "\n").
- No entanto, não é possível utilizar os caracteres curingas, por exemplo o %f, para concatenar as linhas.
- Sintaxe:

puts("Digite a sua mensagem");

Comando "gets"

- Facilita a operação de variáveis string. Ao término da entrada de dados (comando <enter>) o sistema coleta todo o resultado da operação.

- Sintaxe:

gets(<nome da variável string>);

- Exemplo:

```
char nomeAluno[80];
```

```
gets(nomeAluno);
```

Fim de aula!

“Analise e Programe”!