



MONITORIA DE RLA

—

Operações com variáveis
e estruturas condicionais

Operações com variáveis:

Tipos de operadores:

- Atribuição:
- Aritméticos:
- Relacionais:
- Lógicos:
- Atribuição simplificada:
- Operadores de incremento:
- Outros operadores:
vírgula(,)/ponto(./ponto-vírgula(;

Tabela com 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	$z = x \% y$

Atribuição “=”: Usado para passar um valor a uma variável. O valor da direita é passado para a esquerda;
ex:

```
numero=5; //numero recebe 5
```

```
num2 = número; //num2 recebe o mesmo valor de numero
```

Aritméticos: São operadores utilizados para operações matemáticas(Olhe a tabela):

ex:

```
Int x=5+5; // valor de x é 10
```

```
float media= 10/2; // valor de média é 5
```

Importante

No caso de uma divisão ela sempre resultará em um valor de ponto flutuante então certifique-se de que seja uma variável float ou double que receba o valor;

Relacionais: Esses operadores retornam true ou false. São utilizados para saber a relação entre valores;

== : igual a

> : maior que

< : menor que

>= : maior ou igual a

<= : menor ou igual a

!= : diferente de

ex: `_Bool x= 5!=10;`
`printf(" valor de x: %d",x);`

```
int y=10;
If(y < 20){
    printf("10 é menor que 20");
}
```

Importante

Foi adicionada um novo tipo que não tinha sido apresentado `_Bool` ele pode receber apenas dois valores (true ou false) para utilizá-lo você deve incluir a biblioteca `stdbool.h` no topo do código:

```
#include <stdbool.h>
```

Tabela com operadores relacionais

Operador	Significado	Exemplo
>	Maior do que	<code>x > 5</code>
>=	Maior ou igual a	<code>x >= 10</code>
<	Menor do que	<code>x < 5</code>
<=	Menor ou igual a	<code>x <= 10</code>
==	Igual a	<code>x == 0</code>
!=	Diferente de	<code>x != 0</code>

Lógicos: São operadores que verificam mais de uma expressão resultando em apenas um valor booleano;

&&: AND (e)

||: OR (ou)

!: NOT (não)

```
_Bool valido=(5>2)&&(5<6);
```

```
if( (5>3) || (5>10) ){  
printf("Valido");  
}
```

Importante

Lembre-se que os operadores têm uma ordem de precedência.
(Pesquise sobre ou veja no livro indicado)

Outros operadores:

, :vírgula(usado para separar variáveis ou elementos).

. : ponto(usado para referenciar elementos)

; :ponto-vírgula(Indica a finalização de uma instrução)

Tabela com operadores lógicos

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

Tabela com operadores de atribuição simplificada

Operador	Significado	Exemplo		
+=	soma e atribui	x += y	igual	x = x + y
-=	subtrai e atribui	x -= y	igual	x = x - y
*=	multiplica e atribui	x *= y	igual	x = x * y
/=	divide e atribui quociente	x /= y	igual	x = x / y
%=	divide e atribui resto	x %= y	igual	x = x % y
&=	E bit a bit e atribui	x &= y	igual	x = x & y
=	OU bit a bit e atribui	x = y	igual	x = x y
^=	OU exclusivo e atribui	x ^= y	igual	x = x ^ y
<<=	desloca à esquerda e atribui	x <<= y	igual	x = x << y
>>=	desloca à direita e atribui	x >>= y	igual	x = x >> y

Atribuição simplificada: Muitos operadores são sempre usados em conjunto com o operador de atribuição. Para tornar essa tarefa mais simples, a linguagem C permite simplificar algumas expressões, como mostra a Tabela

Tabela com operadores de incremento

Operador	Significado	Exemplo	Resultado
++	incremento	++x ou x++	x = x + 1
--	decremento	--x ou x--	x = x - 1

Incremento: Além dos operadores simplificados, a linguagem C também possui operadores de incremento (++) e decremento (--). Esses operadores podem ser utilizados sempre que for necessário somar uma unidade (incremento) ou subtrair uma unidade (decremento) a determinado valor.

Estruturas condicionais

Estrutura If :

O if é uma estrutura condicional que recebe uma expressão dentro do () que retorna um valor booleano. O if executa um bloco de código caso o valor seja verdadeiro.

Elé é geralmente usado com o else que executa outro bloco caso o valor seja falso.

O if tem algumas variações:

```
If(condição)comando;
```

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

```
If(condição){  
    Sequência de comandos;  
}else{  
    Sequência de comandos;  
}
```

Estrutura Switch:

Switch é uma estrutura que tem a mesma função do if, a diferença é que ela é melhor para ser usada quando se tem uma entrada de valores definidos e que não exige tanto desvio.

```
Switch(condição){  
    case 1:  
        Sequência de comandos  
        break;//o break finaliza  
    case 2:  
        Sequência de comandos  
        break;  
    default:  
        Sequência de Comandos  
        break;  
}
```

PROJETO

Prática: <https://replit.com/@GuilhermeAlme18/Aprendendo-C>

Projeto: https://github.com/Guilfullstack/Task_manager.git



Implementação

Hora de implementar:

Com base no que foi ensinado, vamos incluir no projeto o código que vai executar a sequência de comandos para as opções de Logar e Sair do menu principal.

Desafio

Implemente o código que vai executar a sequência de comandos para a opção de Cadastrar.