



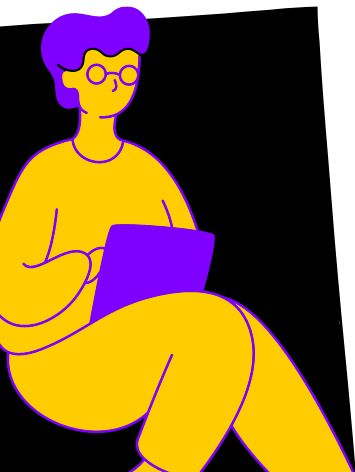
Operadores Lógicos, Tabela Verdade e Técnicas de Programação

PROJETO SIM, ELAS
PODEM!

Operadores Lógicos

Quer combinar condições? Utilize operadores lógicos!
Conheça eles:

! = **NÃO** lógico, operação de negação
&& = **E** lógico, conjunção
|| = **OU** lógico, disjunção



Exemplo

Fazer um programa para identificar se um número é positivo e par.

Condição 1: positivo ou negativo?

Condição 2: par ou ímpar?

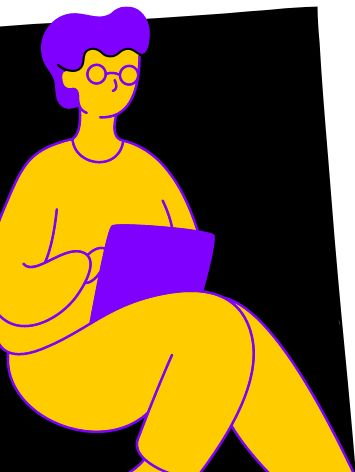
Operadores Lógicos

Representação sem operador lógico

```
if (num >= 0)
    if (num % 2 == 0)
        cout << "Numero par nao negativo." << endl;
```

Representação com operador lógico

```
if ((num>=0) && (num%2 == 0))
    cout << "Numero par nao negativo." << endl;
```

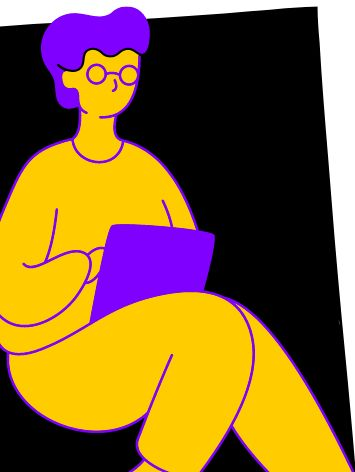


Operadores Lógicos

! é uma expressão que o valor é a negação lógica da expressão dada. Exemplo:

!0 (falso) é 1 (verdadeiro)

!1 (falso) é 0 (verdadeiro)

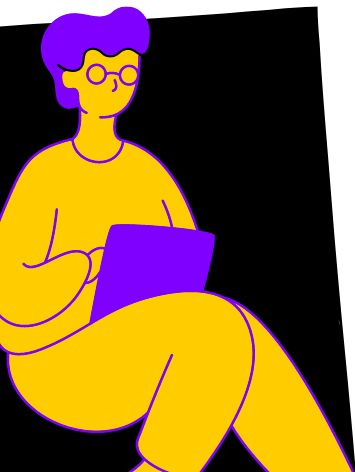


(Condição 1 && Condição 2) = Verdadeiro, se Condição 1 é Verdadeira e Condição 2 é Verdadeira.

(Condição 1 || Condição 2) = Verdadeiro, se Condição 1 é Verdadeira ou Condição 2 é Verdadeira.

Operadores Lógicos

$expr_1$	$expr_2$	$expr_1 \ \&\& \ expr_2$	$expr_1 \ \ expr_2$
<i>verdadeiro</i>	<i>verdadeiro</i>	<i>verdadeiro</i>	<i>verdadeiro</i>
<i>verdadeiro</i>	<i>falso</i>	<i>falso</i>	<i>verdadeiro</i>
<i>falso</i>	<i>verdadeiro</i>	<i>falso</i>	<i>verdadeiro</i>
<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>

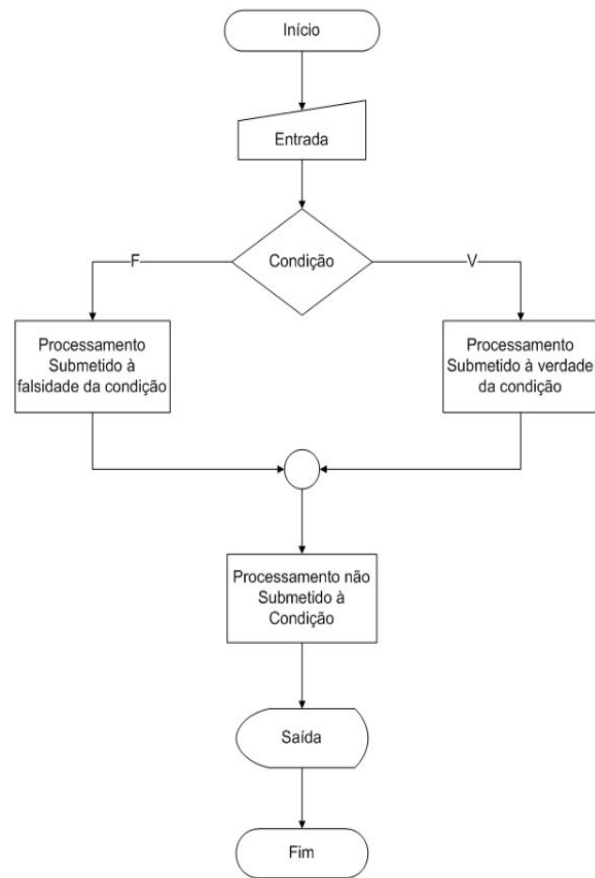




Estrutura Condicional Composta

Estrutura com duas ou mais condições para ser executada

Faça um programa que leia um número inteiro. Caso ele seja positivo e par, imprima “ É par e positivo”





Estrutura Condicional Composta

```
#include <bits/stdc++.h>

using namespace std;

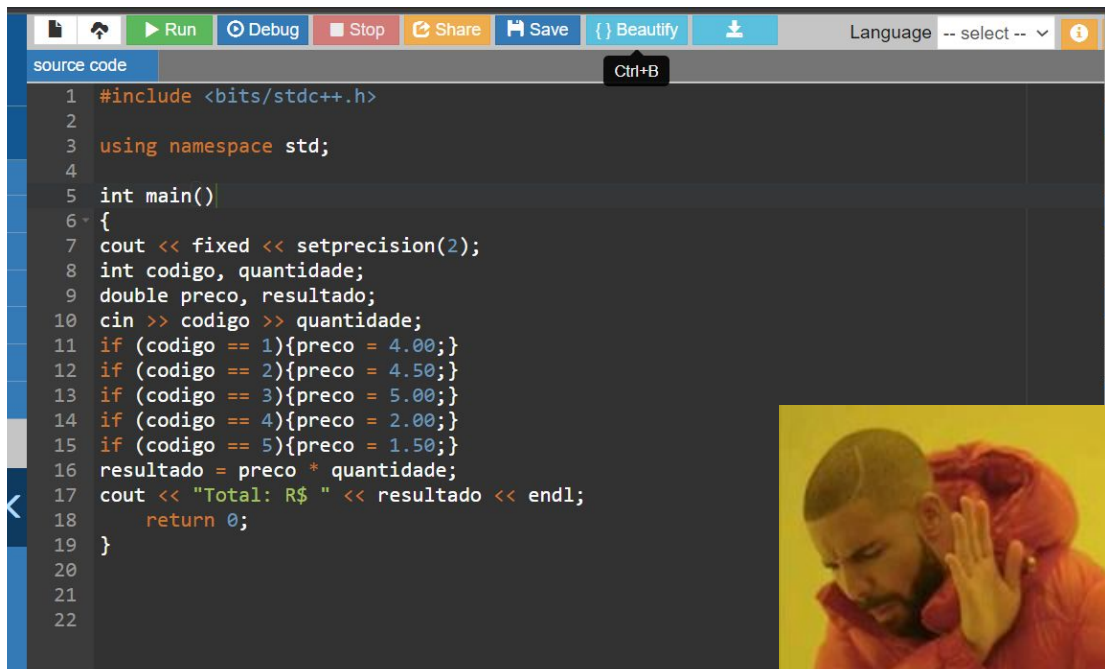
int main()
{
    int a;

    cin >> a;

    if (a>0 && a%2==0){
        cout << "É par e positivo" << endl;
    }
}
```

Indentação

Sem indentação



A screenshot of a code editor window with a dark theme. The editor contains C++ code that is not indented. The code is as follows:

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << fixed << setprecision(2);
8     int codigo, quantidade;
9     double preco, resultado;
10    cin >> codigo >> quantidade;
11    if (codigo == 1){preco = 4.00;}
12    if (codigo == 2){preco = 4.50;}
13    if (codigo == 3){preco = 5.00;}
14    if (codigo == 4){preco = 2.00;}
15    if (codigo == 5){preco = 1.50;}
16    resultado = preco * quantidade;
17    cout << "Total: R$ " << resultado << endl;
18    return 0;
19 }
20
21
22
```

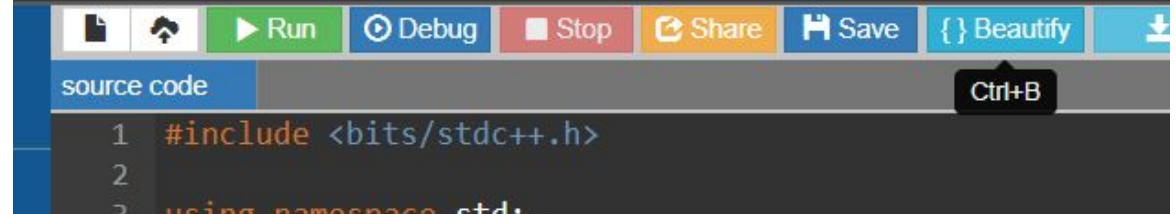
The editor's toolbar at the top includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a Language dropdown menu. A tooltip for 'Ctrl+B' is visible over the Beautify button.



Indentação



www.onlinegdb.com



Indentação

Com indentação

```
source code
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main ()
6 {
7     cout << fixed << setprecision (2);
8     int codigo, quantidade;
9     double preco, resultado;
10    cin >> codigo >> quantidade;
11
12    if (codigo == 1)
13    {
14        preco = 4.00;
15    }
16    if (codigo == 2)
17    {
18        preco = 4.50;
19    }
20    if (codigo == 3)
21    {
22        preco = 5.00;
23    }
24    if (codigo == 4)
25    {
26        preco = 2.00;
27    }
28    if (codigo == 5)
29    {
30        preco = 1.50;
31    }
32    resultado = preco * quantidade;
33    cout << "Total: R$ " << resultado << endl;
34    return 0;
35 }
36
```



Comentários

Em linha



The image shows a screenshot of a C++ IDE. The top toolbar contains buttons for Run, Debug, Stop, Share, Save, Beautify, and a Language dropdown menu. The source code is displayed in a dark-themed editor with line numbers 1 through 10. The code is as follows:

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello World!" << endl;
8     // estou imprimindo "Hello Word"
9
10 }
```

An arrow points from a text box containing the text "somente uma linha de comentário" to the comment line on line 8.



Comentários

Em blocos

w.onlinegdb.com

Run Debug Stop Share Save Beautify Language -- select --

source code

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello World!" << endl;
8     /* início de comentário
9     estou imprimindo "Hello Word"
10    utilizando a linguagem c++
11    */
12 }
13
14 }
```

final de comentário



Se $a=6$, $b=4$, $c=3$ e $d=2$, qual será o resultado de:

1. $(a > b) \ \&\& \ (c == a/d)$
2. $(!(b < d)) \ || \ (a == d*d)$
3. $(c > d) \ \&\& \ (a \% d == 1)$
4. $((d > c) \ \&\& \ (b < a)) \ || \ ((b > c) \ \&\& \ (c > d))$



**Vamos treinar
!?**

1037 - URI

1041 - URI



Intervalo

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Você deve fazer um programa que leia um valor qualquer e apresente uma mensagem dizendo em qual dos seguintes intervalos $[0,25]$, $(25,50]$, $(50,75]$, $(75,100]$ este valor se encontra. Obviamente se o valor não estiver em nenhum destes intervalos, deverá ser impressa a mensagem "Fora de intervalo".

O símbolo $($ representa "maior que". Por exemplo:

$[0,25]$ indica valores entre 0 e 25.0000, inclusive eles.

$(25,50]$ indica valores maiores que 25 Ex: 25.00001 até o valor 50.000000

Entrada

O arquivo de entrada contém um número com ponto flutuante qualquer.

Saída

A saída deve ser uma mensagem conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
25.01	Intervalo (25,50]
25.00	Intervalo [0,25]
100.00	Intervalo (75,100]
-25.02	Fora de intervalo

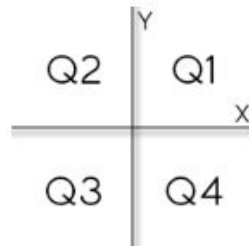


Coordenadas de um Ponto

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Leia 2 valores com uma casa decimal (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem ($x = y = 0$).



Se o ponto estiver na origem, escreva a mensagem “Origem”.

Se o ponto estiver sobre um dos eixos escreva “Eixo X” ou “Eixo Y”, conforme for a situação.

Entrada

A entrada contém as coordenadas de um ponto.

Saída

A saída deve apresentar o quadrante em que o ponto se encontra.

Exemplo de Entrada	Exemplo de Saída
4.5 -2.2	Q4
0.1 0.1	Q1
0.0 0.0	Origem

