

Roteiro 3 - Estrutura Condicional

Uma das tarefas fundamentais de qualquer programa é decidir o que deve ser executado, para isso, temos comandos de decisão que permitem determinar qual a ação deve ser tomada a partir de um resultado de uma expressão condicional. Em C/C++ temos três comandos de decisão:

if
if-else
switch-case

Estrutura Condicional Simples

```
# include <nome da biblioteca>
int main()
{
    if (condição)
    {
        Bloco de comandos;
    }
}
```

- Observe que o **bloco de comandos** só será executado se a **condição** for verdadeira. Uma condição é uma comparação que possui dois valores possíveis (verdadeiro ou falso).
- Necessário a utilização de **chaves** { } quando houver mais de um comando no **Bloco de comandos**. Casos quando o Bloco de comandos possui apenas um comando, não precisa utilizar **chaves**.
- No exemplo acima, o bloco de comandos entre chaves só será executado se a condição for verdadeira; caso a condição seja falsa, a execução do programa passa para o comando seguinte à chave de fechamento do bloco de comandos.

Exemplo: O exemplo identifica qual o maior entre dois números digitados.

```
8  #include <iostream>
9
10 using namespace std;
11 int main()
12 {
13     float num1, num2;
14     cout << "\n Digite o valor do primeiro numero " << endl;
15     cin >> num1;
16     cout << "\n Digite o valor do segundo numero " << endl;
17     cin >> num2;
18     if(num1 > num2)
19     {
20         cout << "\n Primeiro numero e maior! " << endl;
21     }
22     if(num2 > num1)
23     {
24         cout << "\n Segundo numero e maior! " << endl;
25     }
26     if(num2 == num1)
27     {
28         cout << "\n Numeros iguais! " << endl;
29     }
30     cout << "\n";
31     return 0;
32 }
33
```

Note que no exemplo anterior 3 comparações são realizadas, sendo que uma delas é desnecessária, como evitá-la? - A resposta é utilizar a estrutura condicional composta.

Estrutura Condicional Composta

```
#include<bibliotecas>
int main()
{
    if(condicao1) {
        bloco de comandos 1;
    }
    else if(condicao2) {
        bloco de comandos2;
    }
    else{
        bloco de comandos3;
    }
}
```

- Na estrutura condicional composta existe um **bloco de comandos** a ser realizado caso a **condição** testada seja falsa.
- No exemplo acima, se a **condição1** for verdadeira, o **bloco de comandos 1** é executado; **senão** se a **condição2** é verificada, caso a mesma seja verdadeira, executa-se o **bloco de comandos 2**; **senão**, o **bloco de comandos 3** é executado.
- **Obs.: O comando else deve ser sempre precedido de um if.**

Exemplo: Identifica maior entre 2 números digitados utilizando estrutura condicional composta.

```
7
8 | #include <iostream>
9
10 | using namespace std;
11 | int main()
12 | {
13 |     float num1, num2;
14 |     cout << "\n Digite o valor do primeiro numero " << endl;
15 |     cin >> num1;
16 |     cout << "\n Digite o valor do segundo numero " << endl;
17 |     cin >> num2;
18 |     if (num1 > num2)
19 |     {
20 |         cout << "\n Primeiro numero e maior! " << endl;
21 |     }
22 |     else if (num2 > num1)
23 |     {
24 |         cout << "\n Segundo numero e maior! " << endl;
25 |     }
26 |     else
27 |     {
28 |         cout << "\n Numeros iguais! " << endl;
29 |     }
30 |     cout << "\n";
31 |     return 0;
32 | }
33
```



Como as **condições das estruturas condicionais** podem ser compostas por expressões aritméticas, operadores relacionais e operadores lógicos, vamos defini-los a seguir.

Operadores relacionais:

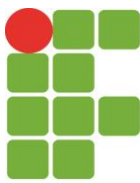
Operador	Exemplo	Comentário
<code>==</code>	<code>x == y</code>	O conteúdo de X é igual ao conteúdo de Y
<code>!=</code>	<code>x != y</code>	O conteúdo de X é diferente do conteúdo de Y
<code><=</code>	<code>x <= y</code>	O conteúdo de X é menor ou igual ao conteúdo de Y
<code>>=</code>	<code>x >= y</code>	O conteúdo de X é maior ou igual ao conteúdo de Y
<code><</code>	<code>x < y</code>	O conteúdo de X é menor que o conteúdo de Y
<code>></code>	<code>x > y</code>	O conteúdo de X é maior que o conteúdo de Y

Operadores lógicos:

Operador	Exemplo	Comentário
<code>&&</code>	<code>if (x >= 3 && x < 7)</code>	Significa que a condição verificada será <code>X >= 3 E X < 7</code> , ou seja, a condição será verdadeira se X estiver entre 3 e 7
<code> </code>	<code>if (x >= 3 x >= 7)</code>	Significa que a condição verificada será <code>X >= 3 OU X <= 7</code> , ou seja, tanto se X for maior que 3 como se X for maior que 7 a condição será verdadeira
<code>!</code>	<code>!x</code>	A <code>!</code> (NÃO) é a <i>negação</i> , ou seja, muda um valor lógico de verdadeiro para falso e de falso para verdadeiro - Resulta 1 (verdadeiro) somente se x for falsa. - Resulta 0 (falso) somente se x for verdadeiro.

Tabela verdade:

Tabela E (<code>&&</code>)	Tabela OU (<code> </code>)	Tabela NÃO (<code>!</code>)
<code>V e V = V</code>	<code>V ou V = V</code>	<code>Não V = F</code>
<code>V e F = F</code>	<code>V ou F = V</code>	<code>Não F = V</code>
<code>F e V = F</code>	<code>F ou V = V</code>	
<code>F e F = F</code>	<code>F ou F = F</code>	

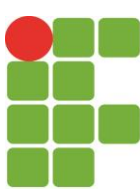


Estrutura switch-case

O comando **switch** é uma estrutura condicional de decisão múltipla que testa o valor de uma expressão e desvia o fluxo de execução do programa de acordo com o resultado do teste.

```
#include<bibliotecas>
int main()
{
    switch (variavel){
        case valor1:
            Bloco de comandos 1;
            break;
        case valor2:
            Bloco de comandos 2;
            break;
        .....
        default: bloco de comandos 3;
    }
}
```

- Em situações mutuamente exclusivas, isto é, se uma situação for executada, as demais **não serão** executadas, um comando seletivo é indicado (estrutura CASE).
- O comando **switch(variável)** avalia o valor da variável para decidir qual case será executado.
- Cada case está associado a UM possível valor da variável, que deve ser obrigatoriamente, do tipo **char** ou **int**.
- O comando **break;** deve ser utilizado para impedir a execução dos comandos definidos nos **cases subsequentes**.
- O comando **default:** é executado quando o valor da variável **não** coincidir com nenhum dos especificados nos *cases*.



Exemplos

1 - Digite e compile os códigos dos exemplos abaixo colocando comentários nas partes que você considera importante.

Exemplo1: Identifica intervalos

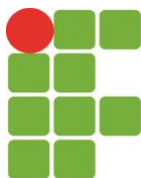
```
4  Entrada: Numeros.
5  Saida: Maior número.
6  */
7
8  #include <iostream>
9
10 using namespace std;
11 int main()
12 {
13     float num1;
14     cout << "\n Digite um numero " << endl;
15     cin >> num1;
16     if(num1 < 0)
17     {
18         cout << "\n Numero negativo! " << endl;
19     }
20     else if(num1 >= 0 && num1 < 7)
21     {
22         cout << "\n Numero esta no intervalo entre 0 e 7 " << endl;
23     }
24     else
25     {
26         cout << "\n Numero maior que 7! " << endl;
27     }
28     cout << "\n";
29     return 0;
30 }
31
```

Exemplo 2: Identifica valores.

```
6  */
7
8  #include <iostream>
9
10 using namespace std;
11 int main()
12 {
13     int num1;
14     cout << "\n Digite um numero " << endl;
15     cin >> num1;
16
17     switch(num1)
18     {
19         case 0: cout << "\n Valor digitado: ZERO! ";
20                 break;
21         case 1: cout << "\n Valor digitado: UM! ";
22                 break;
23         case 2: cout << "\n Valor digitado: DOIS! ";
24                 break;
25         case 3: cout << "\n Valor digitado: TRES! ";
26                 break;
27         default: cout << "\n Valor digitado: diferente de 0, 1, 2 ou 3 ";
28                 break;
29     }
30     cout << "\n";
31     return 0;
32 }
33
```

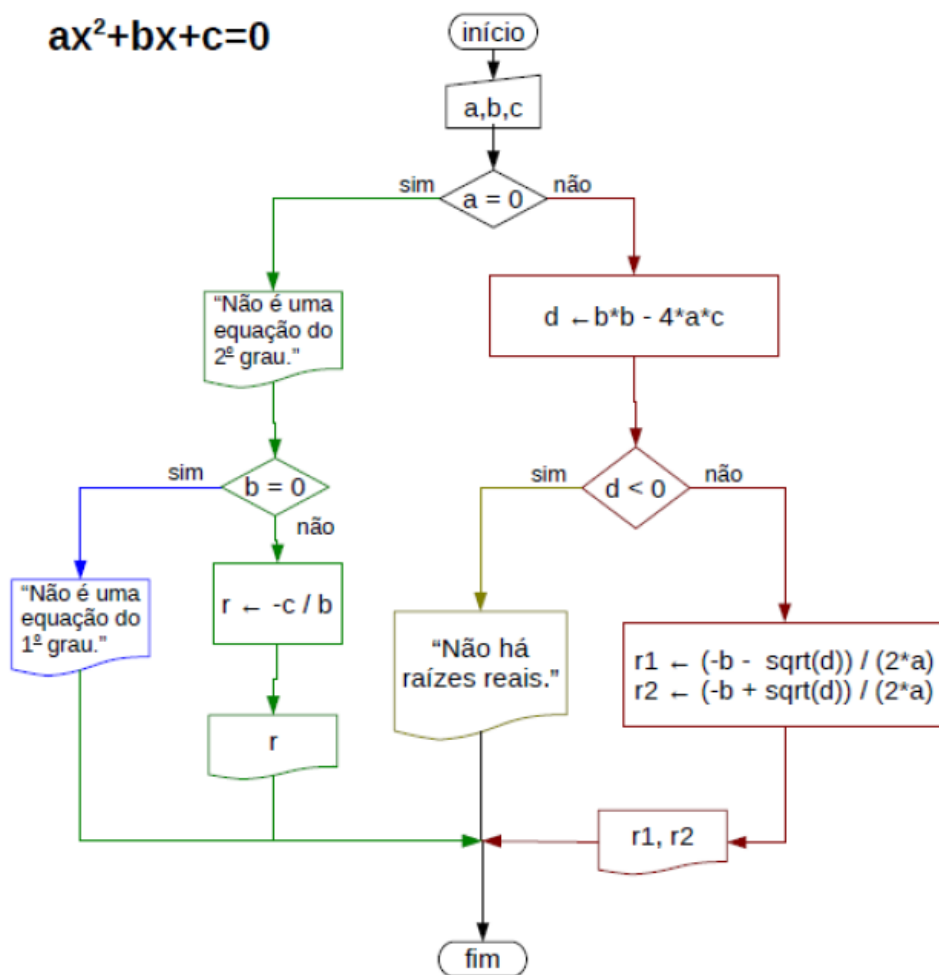
Exemplo 3: Escolha entre opções

```
12  int main()  
13  {  
14      char opcao;  
15      int num1, num2, num3;  
16      cout<<"Menu de opcoes: ";  
17      cout<<"\n a - Para solicitar um numero ";  
18      cout<<"\n b - Para solicitar dois numeros";  
19      cout<<"\n c - Para solicitar tres numeros \n ";  
20      cin>>opcao;  
21      switch(opcao)  
22      {  
23          case 'a':  
24              cout<<"\n Opcao escolhida foi a primeira";  
25              cout<<"\n Digite um numero: ";  
26              cin>>num1;  
27              break;  
28          case 'b':  
29              cout<<"\n Opcao escolhida foi a segunda";  
30              cout<<"\n Digite o primeiro numero: ";  
31              cin>>num1;  
32              cout<<"\n Digite o segundo numero: ";  
33              cin>>num2;  
34              break;  
35          case 'c':  
36              cout<<"\n Opcao escolhida foi a terceira";  
37              cout<<"\n Digite o primeiro numero: ";  
38              cin>>num1;  
39              cout<<"\n Digite o segundo numero: ";  
40              cin>>num2;  
41              cout<<"\n Digite o terceiro numero: ";  
42              cin>>num3;  
43              break;  
44          default: cout<<"\n Opcao invalida!";  
45              break;  
46      }  
47      cout<<"\nFim do programa";  
48      cout<<"\n";  
49      return 0;  
50  }
```

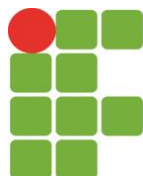


Exercícios

1. Faça um programa que identifique se um número digitado pelo usuário é positivo, negativo ou zero.
2. Faça um algoritmo que dado um número digitado pelo usuário mostre se o mesmo é par ou ímpar.
3. Crie um programa que solicite a entrada de dois números, em seguida imprima na tela o quadrado do menor e a raiz quadrada do maior número, caso os números sejam iguais mostre uma mensagem informando que os números são iguais. Lembre-se de verificar as condições para um número possuir raiz quadrada real.
4. Refaça o exercício 11 do roteiro anterior considerando todas as condições possíveis, conforme fluxograma abaixo.



5. O custo de um carro novo ao consumidor é a soma do custo de fábrica, a porcentagem do distribuidor e os impostos. O valor pago pela distribuição e dos impostos é calculado com base no custo de fábrica. As porcentagens estão apresentadas na tabela abaixo. Faça um programa que receba o custo de fábrica de um carro e mostre o custo ao consumidor



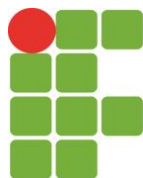
Custo de Fábrica	% Distribuidor	% Impostos
Até R\$ 28000.00	5	Isento
Entre R\$28000.01 e R\$45000.00	10	15
Acima de R\$45000.00	15	20

6. Faça um programa que simule uma calculadora simples. Para isso, o usuário deverá entrar com dois números quaisquer e a operação que deseja realizar: '+' soma; '-' subtração; '*' multiplicação e '/' divisão. Lembre-se das condições para realizar a divisão. Utilize a estrutura **switch/case** e apresente um menu de opções para o usuário.
7. Faça um programa que solicite ao usuário que digite um caracter e em seguida imprima se o caracter digitado é ou não uma vogal.
8. Uma empresa decide dar um aumento de 30% aos funcionários com salário inferior à R\$700,00 e 10% aos funcionários com salário superior ou igual à R\$700,00. Faça um algoritmo que calcule o reajuste e mostre o novo salário.
9. Na cidade de Sabará, para transferências de veículos, a SABATRAN cobra uma taxa de 1% para carros fabricados antes de 1990 e uma taxa de 1.5% para os fabricados de 1990 em diante. Essa taxa está incidindo sobre o valor de tabela do carro. Implemente um algoritmo que lê o ano de fabricação e o preço do carro e a seguir calcula e imprime imposto a ser pago.
10. Criar um algoritmo que a partir da idade e peso do paciente calcule a dosagem de determinado medicamento e **imprima a receita informando quantas gotas do medicamento** o paciente deve tomar por dose. Considere que o medicamento em questão possui 500 mg por ml, e que cada ml corresponde a 20 gotas.
 - Adultos ou adolescentes desde 12 anos, inclusive, se tiverem peso igual ou acima de 60 quilos devem tomar 1000 mg; com peso abaixo de 60 quilos devem tomar 875 mg.
 - Para crianças e adolescentes abaixo de 12 anos a dosagem é calculada pela peso corpóreo conforme a tabela a seguir:

Peso	Dosagem
5 kg a 9 kg	125 mg
9.1 kg a 16 kg	250 mg
16.1 kg a 24 kg	375 mg
24.1 kg a 30 kg	500 mg
Acima de 30 kg	750 mg

11. Crie um algoritmo que receba o valor de x , calcule e imprima o valor de $f(x)$.

$$f(x) = \begin{cases} 1, & \text{se } x \leq 1 \\ 2, & \text{se } 1 < x \leq 2 \\ x^2, & \text{se } 2 < x \leq 3 \\ x^3, & \text{se } x > 3 \end{cases}$$



12. Faça um algoritmo que receba a idade de um nadador e mostre a categoria do mesmo. Faça os testes necessários em relação a idade inválida.

Categoria	Operação
Infantil	5 a 7 (inclusive)
Juvenil	8 a 10 (inclusive)
Adolescente	11 a 15 (inclusive)
Adulto	16 a 40 (inclusive)
Sênior	Acima de 40 (inclusive)

13. O IMC (Índice de Massa Corporal) é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é $IMC = PESO / (ALTURA^2)$. Elabore um algoritmo que leia o peso e a altura de um adulto e mostre sua condição.

IMC em adultos	Condição
abaixo de 18.5	abaixo do peso
entre 18.5 e 25	peso normal
entre 25 e 30	acima do peso
acima de 30	Obeso

14. Elabore um programa que calcule e exiba a tensão S de uma barra cilíndrica de diâmetro D submetida a uma carga Q . Os valores de D e Q devem ser fornecidos pelo usuário do programa via teclado. Utilize a fórmula:

$$S = \frac{4 * Q}{\pi * D^2} * n$$

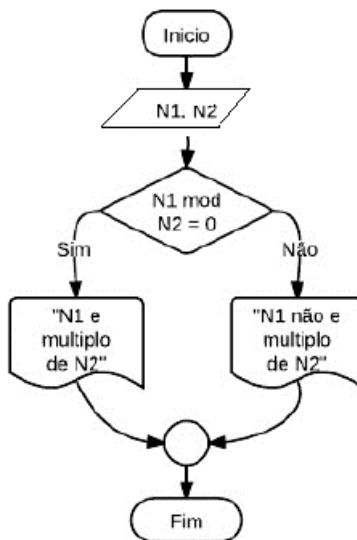
onde, $n = 2$ para $D > 100$; $n = 6$ para $D < 50$ e; $n = 4$ para outros valores de D .

15. Faça um programa que receba de entrada o ano atual e o ano de nascimento de uma pessoa e verifique se essa pessoa não pode votar, é obrigada a votar ou o voto é facultativo, segundo as regras brasileiras.

Idade	Voto
Menor de 16	Não vota
16 à 18 e acima de 70	Facultativo
Entre 18 e 70	Obrigatório

16. Faça um programa que recebe de entrada três valores. Verificar se esses valores podem formar um triângulo. Caso não, informar que não forma um triângulo; caso sim, imprima que tipo de triângulo é formado (escaleno, isósceles ou equilátero).
17. Implemente um programa que receba um número inteiro de 3 dígitos de entrada. Após isso, verifique se o mesmo é palíndromo, ou seja, é o mesmo número lido da direita para esquerda ou vice-versa. Ex.: 121, 353, etc.

18. Elabore um programa que realize o mesmo procedimento do fluxograma abaixo.



19. Implemente um programa que receba um número inteiro positivo representando um ano qualquer e imprima se o ano é bissexto ou não. Um ano bissexto é definido quando: (ano % 4 = 0 e (ano % 100 != 0 ou ano % 400 == 0)).

20. Faça um algoritmo que receba três números e mostre-os em ordem crescente e decrescente.

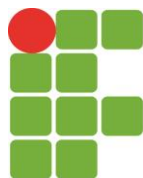
21. Faça um algoritmo que receba duas datas (dia, mês e ano) e determine a maior entre elas.

22. O número de inscrição no CPF é composto de onze dígitos decimais, sendo os oito primeiros aleatoriamente designados no momento da inscrição. Já o nono (antepenúltimo) dígito indica a região fiscal responsável pela inscrição (MG é a região 6, portanto em todos CPF's emitidos em MG o nono dígito é 6). Por fim, o décimo e o décimo primeiro são **dígitos verificadores calculados de acordo com um algoritmo definido pela Receita Federal** e publicamente conhecido. Assim sendo, seja **D** os nove primeiros dígitos de um número de CPF qualquer visitado da direita para a esquerda, na forma: $D = (d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8)$, onde d_i representa o dígito do CPF na posição i e d_0 representa a posição mais a direita deste CPF. Ou seja, os dígitos do CPF 123456789 são representados por $D = (9, 8, 7, 6, 5, 4, 3, 2, 1)$. Diante disso, os dígitos verificadores **v1** e **v2** podem ser calculados pelas expressões:

$$v_1 = \left[\left(\sum_{i=0}^{n-1} d_i \times [9 - (i \bmod 10)] \right) \bmod 11 \right] \bmod 10$$

$$v_2 = \left\{ \left[\left(\sum_{i=0}^{n-1} d_i \times \{9 - [(i+1) \bmod 10]\} \right) + (v_1 \times 9) \right] \bmod 11 \right\} \bmod 10$$

Assim, implemente um algoritmo que receba de entrada o valor dos 9 primeiros dígitos de um CPF (uma única variável inteira), e dos 2 dígitos verificadores. Desmembre os 9 dígitos do CPF utilizando as operações de divisão e resto da divisão inteira. Faça os cálculos, de acordo com a expressão acima, para verificar se os dígitos verificadores do CPF digitado são válidos. Como exemplo, utilizem o valor inicial dos CPF 111444777, os dígitos verificadores devem ser $v_1 = 3$ e $v_2 = 5$. Ou utilizem o próprio CPF como exemplo.



23. Faça um programa que recebe de entrada o código de um produto e a quantidade comprada do mesmo. Calcule e mostre:

- O preço unitário do produto, segundo Tabela I
- Preço total da nota
- O valor do desconto, segundo Tabela II, aplicado sobre o valor total da nota
- Preço final ao consumidor após desconto
- Não se esqueça de verificar a existência do produto

Código	Valor unitário
1 a 10	R\$10,00
11 a 20	R\$20,00
21 a 30	R\$30,00
31 a 40	R\$40,00

Total da nota	% Desconto
Até R\$ 250	5%
Entre R\$250 e R\$500	10%
Acima de R\$500	15%

24. Um retângulo é dito de ouro quando a razão entre o lado maior e o lado menor é igual ao número de ouro 1.618. Implemente um algoritmo que dados os lados de um retângulo, determine se o mesmo é ou não um retângulo de ouro.

25. Faça um algoritmo que receba de entrada a altura e o sexo de uma pessoa ('M' - Masculino ou 'F' - Feminino), construa um algoritmo que calcule o peso ideal da pessoa de acordo com as regras:

- Homens: $(72.7 * \text{altura}) - 58$

- Mulher: $(62,1 * \text{altura}) - 44.7$

26. Faça um algoritmo que receba o salário base e o tempo de serviço de um funcionário. Calcule e mostre:

(i) O imposto, conforme tabela:

SALÁRIO BASE	% SOBRE O SALÁRIO BASE
< R\$ 200,00	Isento
Entre R\$ 200,00 (inclusive) e R\$ 450,00 (inclusive)	3%
Entre R\$ 450,00 e R\$ 700,00	8%
\geq R\$ 700,00	12%

(ii) A gratificação, conforme tabela:

SALÁRIO BASE	TEMPO DE SERVIÇO	GRATIFICAÇÃO
Superior a R\$ 500,00	Até 3 anos	20
	Mais de 3 anos	30
Até R\$ 500,00	Até 3 anos	23
	Entre 3 e 6 anos	35
	De 6 anos para cima	33

(iii) O salário líquido, ou seja, salário base menos imposto mais gratificação

(iv) A categoria que está na tabela a seguir:

SALÁRIO LÍQUIDO	CLASSIFICAÇÃO
Até R\$ 350,00	A
Entre R\$ 350,00 e R\$ 600,00	B
De R\$ 600,00 para cima	C