

[◀ VOLTAR](#)

Implementação de desvios condicionais em C

Apresentar os conceitos sobre desvios condicionais simples, composto, encadeado e de múltipla escolha na linguagem C, bem como exemplos das suas aplicações.

NESTE TÓPICO

- › Introdução
- › Desvio Condicional Simples em C
- › Exemplo 1
- › Exemplo 2
- › Desvio Condicional Composto em C

Marcar
tópico



Introdução

Os desvios condicionais, também chamados de estruturas de seleção ou decisão, são utilizados quando existe a necessidade de verificar condições para a realização de uma instrução ou de uma sequência de instruções. Os testes de seleção também podem ser utilizados para verificar opções de escolha. A tomada de decisão sempre deve estar de acordo com o resultado de uma expressão lógica.

Os comandos de decisão ou desvio fazem parte das técnicas de programação que conduzem às estruturas de programas que não são totalmente sequenciais. As estruturas de decisão (condicionais) são utilizadas para tomar uma decisão baseada no resultado da avaliação de uma condição de controle e seleciona uma ou mais ações possíveis (comandos) para serem executados pelo computador.

Com as instruções de desvio pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores. As principais estruturas de decisão são: simples, composto, encadeado e o de múltipla escolha. Neste tópico serão abordados conceitos sobre cada desvio com exemplos.



Desvio Condicional Simples em C

São utilizadas para verificar se dada condição é atendida: se for, um conjunto de instruções deverá ser executado; se não for, o fluxo da execução do algoritmo seguirá após o fim do bloco de decisão.

Na linguagem C, é utilizada a estrutura de decisão "if" que, normalmente, vem acompanhada de uma ou mais linhas de instrução, ou seja, se determinada condição for satisfeita (verdadeira) pelo comando "if", então uma ou mais linhas de instrução serão executadas.

O comando "if" é equivalente ao comando "se" em Português Estruturado, e deve ser utilizado da seguinte forma:

```
1. if ( <condicao> ) {  
2.  
3.     <instrucoes para condicao verdadeira>  
4.  
5. }  
6.
```

IMPORTANTE

- A condição deve ser uma expressão lógica.
- Quando a condição for Verdadeira, serão executadas todas as instruções definidas naquele bloco.
- Quando a condição for Falsa, não serão executadas as instruções definidas naquele bloco.
- Quando houver somente uma instrução a ser executada caso a condição seja verdadeira, pode-se omitir as chaves que delimitam o bloco de instruções. Porém, se mais de uma linha de instrução estiver definida no bloco, então as chaves para a delimitação do bloco deverão ser, obrigatoriamente, colocadas.

No trecho de código a seguir, é mostrado um exemplo da utilização do desvio simples ("if") no qual somente uma linha de instrução deverá ser executada caso a condição seja verdadeira.

```
1. if (x > 0)  
2.     printf ("O valor de x eh maior do que zero! \n");
```

No trecho de código a seguir, é mostrado um exemplo da utilização do desvio simples ("if") no qual mais de uma linha de instrução deverá ser executada caso a condição seja verdadeira. Neste caso, a colocação das chaves é obrigatória.



```
1.  if (x != 0){
2.      res = x + 5;
3.      printf ("Resultado = %d", res);
4.  }
```

Exemplo 1

No exemplo a seguir, é feita a leitura de dois valores numéricos inteiros, denominados A e B. Em seguida, é efetuada a adição entre eles e o resultado dessa operação só é exibido caso seu valor seja maior que 10.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.  main (){
5.      int A, B, X;
6.      printf ("\n Digite os valores de A e B:");
7.      scanf ("%d%d", &A, &B);
8.      X = A + B;
9.
10.     if (X > 10)
11.         printf ("\n Valor de X: %d", X);
12.
13.     system ("PAUSE");
14. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 1 passo a passo.

SIMULAÇÃO

Exemplo 1 – Desvio condicional simples

```
#include <stdio.h>
#include <stdlib.h>
```

```
main (){
```

```
    int A, B, X;
```

```
    printf ("\n Digite os valores de A e B:");
```

```
    scanf ("%d%d", &A, &B);
```

```
    X = A + B;
```

```
    if (X > 10)
```

```
        printf ("\n Valor de X: %d", X);
```

```
    system ("PAUSE");
```

```
}
```

Área de execução do programa

Digite os valores de A e B:

Simulação da execução do desvio condicional simples em C - Exemplo 1



Exemplo 2

No exemplo a seguir, é feita a leitura do salário de um funcionário. Em seguida, caso o salário informado seja menor ou igual a 2000, então é feito o acréscimo de 30% sobre seu valor. Por fim, o novo salário é exibido.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. main (){
5.     float salario, novo_salario;
6.     printf ("\n Digite o salario:");
7.     scanf ("%f", &salario);
8.
9.     if (salario <= 2000)
10.        novo_salario = salario * 1.30;
11.
12.     printf ("\n Novo salario=%.2f \n", novo_salario);
13.     system ("PAUSE");
14. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 2 passo a passo.

SIMULAÇÃO

Exemplo 2 – Desvio Condicional Simples

```
#include <stdio.h>
#include <stdlib.h>

main (){
float salario, novo_salario;
printf ("\n Digite o salario:");
scanf ("%f", &salario);

if (salario <= 2000)

    novo_salario = salario * 1.30;

printf ("\n Novo salario=%.2f \n", novo_salario);
system ("PAUSE");

}
```

Área de execução do programa

Digite o salario:

Simulação da execução do desvio condicional simples em C - Exemplo 2

Fonte:

Desvio Condicional Composto em C

O desvio condicional composto prevê uma condição com 2 conjuntos de instruções para serem realizados de acordo com a avaliação da resposta: um bloco de instruções para resposta verdadeiro e um bloco de instruções para resposta falso.



Na linguagem C, é utilizada a estrutura de decisão “if...else”, que funciona exatamente como a estrutura “if”, com apenas a diferença que sempre alguns dos blocos serão executados.

O comando "if...else" é equivalente ao comando "se...entao...senao" da linguagem algorítmica, e deve ser utilizado da seguinte forma:

```
1.  if ( <condicao> ) {  
2.  
3.      <instrucoes para condicao verdadeira>  
4.  
5.  }  
6.  
7.  else {  
8.  
9.      <instrucoes para condicao falsa>  
10.  
11. }
```

Exemplo 3

Neste exemplo é feita a leitura de dois valores numéricos inteiros. Caso a adição entre eles resulte em um valor maior ou igual a 10, será atribuído a uma nova variável o valor resultante somado a 5. Caso contrário (adição menor que 10), será atribuído a uma nova variável o valor resultante subtraído de 7. Por fim, o valor da nova variável será exibido.

```
1.  #include <stdio.h>  
2.  #include <stdlib.h>  
3.  
4.  main (){  
5.      int A, B, X, R;  
6.      printf ("\n Digite os valores de A e B:");  
7.      scanf ("%d%d", &A, &B);  
8.      X = A + B;  
9.  
10.     if (X > 10){  
11.         R = X + 5;  
12.     }  
13.     else {  
14.         R = X - 7;  
15.     }  
16.  
17.     printf ("\n Valor de R:%d", R);  
18.     system ("PAUSE");  
19. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 3 passo a passo.





Simulação da execução do desvio condicional composto em C - Exemplo 3

Exemplo 4

No exemplo a seguir, é feita a leitura do salário de um funcionário. Em seguida, caso o salário informado seja menor ou igual a 2000, então é feito o acréscimo de 30% sobre seu valor. Caso contrário (salário maior do que 2000), será feito um acréscimo de 10% sobre seu valor. Por fim, o novo salário é exibido.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. main (){
5.     float salario, novo_salario;
6.     printf ("\n Digite o salario:");
7.     scanf ("%f", &salario);
8.
9.     if (salario <= 2000) {
10.         novo_salario = salario * 1.30;
11.     }
12.     else {
13.         novo_salario = salario * 1.10;
14.     }
15.
16.     printf ("\n Novo salario:%.2f", novo_salario);
17.     system ("PAUSE");
18. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 4 passo a passo.



< 08 / 01 >

Simulação da execução do desvio condicional composto em C - Exemplo 4

Desvio Condicional Encadeado em C

Um desvio condicional encadeado é uma sequência de testes de seleção, os quais serão executados ou não de acordo com o resultado das condições e de acordo com o encadeamento dos testes.

Existem casos em que é necessário estabelecer algumas verificações lógicas de condições definidas sucessivamente. A ideia aqui apresentada sugere a possibilidade de usar uma condição dentro de outra condição, o que leva a uma estrutura de decisão encadeada ou aninhada.

O desvio condicional encadeado em C deve ser utilizado da seguinte forma:



```
1.  if ( <condicao1> ) {
2.
3.      <instrucoes para condicao1 verdadeira>
4.
5.  }
6.
7.  else {
8.
9.      if ( <condicao2> ) {
10.
11.          <instrucoes para condicao2 verdadeira>
12.
13.      }
14.      else {
15.
16.          <instrucoes para condicao2 falsa>
17.
18.      }
19.
20. }
```

Exemplo 5

Neste exemplo é efetuado o cálculo do reajuste do salário de um funcionário. Considere que o funcionário deve receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%; caso seja ainda maior que 1000, o reajuste deverá ser de 5%. No final do algoritmo, o valor do novo salário é exibido.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.  main (){
5.      float salario, novo_salario;
6.      printf ("\n Digite o salario:");
7.      scanf ("%f", &salario);
8.
9.      if (salario < 500) {
10.          novo_salario = salario * 1.15;
11.      }
12.      else {
13.          if (salario <= 1000) {
14.              novo_salario = salario * 1.10;
15.          }
16.          else {
17.              novo_salario = salario * 1.05;
18.          }
19.      }
20.
21.      printf ("\n Novo salario:%.2f", novo_salario);
22.      system ("PAUSE");
23. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 5 passo a passo.



《14 / 01》

Simulação da execução do desvio condicional encadeado em C - Exemplo 5

Fonte:

Exemplo 6

Neste exemplo é feita uma operação sobre um número inteiro de acordo com o seu sinal (positivo, negativo ou igual a zero). Caso o número seja positivo, será adicionado 5 ao seu valor; se ele for negativo, será multiplicado 7 ao seu valor e, caso seja igual a zero, será subtraído 3 ao seu valor. Por fim, o número inteiro atualizado será exibido.



```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.  main (){
5.      int x;
6.      printf ("\n Digite o valor de x:");
7.      scanf ("%d", &x);
8.
9.      if (x > 0) {
10.         x = x + 5;
11.     }
12.     else {
13.         if (x < 0) {
14.             x = x * 7;
15.         }
16.         else {
17.             x = x - 3;
18.         }
19.     }
20.     printf ("\n Novo Valor de x:%d", x);
21.     system ("PAUSE");
22. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 6 passo a passo.

SIMULAÇÃO

Exemplo 6 – Desvio condicional encadeado em C

```
#include <stdio.h>
#include <stdlib.h>
```

```
main (){
    int x;
    printf ("\n Digite o valor de x:");
    scanf ("%d", &x);
    if (x > 0){
        x = x + 5;
    }
    else {
        if (x < 0) {
            x = x * 7;
        }
        else {
            x = x - 3;
        }
    }
    printf ("\n Novo Valor de x:%d", x);
    system ("PAUSE");
}
```

Área de execução do programa

Digite o valor de x:
x = 6

Simulação da execução do desvio condicional encadeado em C - Exemplo 6

Fonte: Elaborado pelo professor conteudista.



Desvio Condicional com Múltipla Escolha em C

Um desvio condicional com múltipla escolha é uma estrutura de seleção que funciona como um conjunto de opções para escolha. Deve ser utilizado quando houver a necessidade de construir um programa no qual seja necessário utilizar uma sequência grande de instruções do tipo "if".

Na linguagem C é utilizada instrução *switch*, que permite avaliar uma expressão entre parênteses, após a palavra *switch*, e compara seu valor com os rótulos dos casos (cases), que são as possibilidades que a expressão pode assumir. A expressão entre parênteses deve ser de valor inteiro ou caractere.

Cada caso deve ser rotulado por uma constante do tipo inteiro ou caractere ou por uma expressão constante. Não é possível utilizar uma variável e nem uma expressão lógica para rótulo. Esta constante deve ser terminada por dois pontos (:) e não por ponto-e-vírgula.

Pode haver uma ou mais instruções seguindo cada *case*. Estas instruções não necessitam estar entre chaves.

O corpo de um *switch* deve estar envolto por chaves.

Se um caso for igual ao valor da expressão, a execução começa nele. Se nenhum caso for satisfeito e existir um caso *default*, a execução começará nele, senão o programa processará as instruções seguintes ao bloco *switch*. Um *default* é opcional.

Os rótulos dos casos devem ser todos diferentes.

O comando *break* garante a execução de apenas uma chamada dentro do switch, ou seja, faz com que o switch seja interrompido assim que aquele bloco for executado. Ele é colocado antes da próxima condição (case) e apenas os comandos existentes da condição selecionada até o comando *break* serão executados.

A instrução *switch* deve ser utilizada da seguinte forma:

```
1. switch ( <expressao> ) {  
2.  
3.     case constante1:  
4.         instruções;  
5.     case constante2:  
6.         instruções;  
7.         ....  
8.     default:          /*opcional*/  
9.         instruções;   /*opcional*/  
10.  
11. }
```

Exemplo 7

No exemplo a seguir, é feita a leitura do código de um produto conforme a tabela. A partir da escolha do código, é feita a exibição do nome do produto.



| Código do Produto | Nome do produto |
|-------------------|-----------------|
| 001 | Caderno |
| 002 | Lápis |
| 003 | Borracha |

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. main (){
5.     int codigo;
6.     printf ("\n Digite o codigo:");
7.     scanf ("%d", &codigo);
8.
9.     switch (codigo) {
10.
11.         case 001:
12.             printf ("\n Caderno \n");
13.             break;
14.         case 002:
15.             printf ("\n Lapis \n");
16.             break;
17.         case 003:
18.             printf ("\n Borracha \n");
19.             break;
20.         default:
21.             printf ("\nCodigo nao encontrado! \n");
22.     }
23.
24.     system ("PAUSE");
25. }

```

Na animação a seguir, é mostrada a simulação da execução do exemplo 7 passo a passo.

| SIMULAÇÃO Exemplo 7 – Desvio condicional com múltipla escolha | |
|--|--|
| <pre> #include <stdio.h> #include <stdlib.h> main (){ int codigo; printf ("\n Digite o codigo:"); scanf ("%d", &codigo); switch (codigo) { case 001: printf ("\n Caderno \n"); break; case 002: printf ("\n Lapis \n"); break; case 003: printf ("\n Borracha \n"); break; default: printf ("\n Nao encontrado! \n"); } system ("PAUSE"); } </pre> | <p>Área de execução do programa</p> <p>Digite o codigo: codigo = 002</p> |



Exemplo 8

No exemplo a seguir, é feita a leitura do código de origem e do preço de um determinado produto. A partir da escolha do código, é feita a exibição do preço junto com sua procedência de acordo com a tabela. Caso o código não seja nenhum dos especificados, o produto deve ser encarado como importado.

Neste exemplo é possível observar que algumas procedências (Nordeste, Sudeste, Centro-Oeste e Nordeste) possuem vários códigos de origem.

| Código de Origem | Procedência |
|------------------|--------------|
| 1 | Sul |
| 2 | Norte |
| 3 | Leste |
| 4 | Oeste |
| 5 ou 6 | Nordeste |
| 7, 8 ou 9 | Sudeste |
| 10 | Centro-Oeste |



```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.  main (){
5.      int codigo;
6.      float preco;
7.      printf ("\n Digite o codigo:");
8.      scanf ("%d", &codigo);
9.      printf ("\n Digite o preco:");
10.     scanf ("%f", &preco);
11.
12.     switch (codigo) {
13.
14.         case 1:
15.             printf ("\n Procedencia Sul - Preco: %.2f \n", preco);
16.             break;
17.         case 2:
18.             printf ("\n Procedencia Norte - Preco: %.2f \n", preco);
19.             break;
20.         case 3:
21.             printf ("\n Procedencia Leste - Preco: %.2f \n", preco);
22.             break;
23.         case 4:
24.             printf ("\n Procedencia Oeste - Preco: %.2f \n", preco);
25.             break;
26.         case 5:
27.         case 6:
28.             printf ("\n Procedencia Nordeste - Preco: %.2f \n", preco);
29.             break;
30.         case 7:
31.         case 8:
32.         case 9:
33.             printf ("\n Procedencia Sudeste - Preco: %.2f \n", preco);
34.             break;
35.         case 10:
36.             printf ("\n Procedencia Centro-Oeste - Preco: %.2f \n", preco);
37.             break;
38.         default:
39.             printf ("\nCodigo nao encontrado! \n");
40.     }
41.
42.     system ("PAUSE");
43. }
```

Na animação a seguir, é mostrada a simulação da execução do exemplo 8 passo a passo.



SIMULAÇÃO

Exemplo 8 – Desvio condicional com múltipla escolha em C

```
#include <stdio.h>
#include <stdlib.h>
```

```
main (){
int codigo;
float preco;
printf ("\n Digite o codigo:");
scanf ("%d", &codigo);
printf ("\n Digite o preco:");
scanf ("%f", &preco);
```

```
switch (codigo) {

case 1:
printf ("\n Procedencia Sul - Preco: %.2f \n", preco);
break;
case 2:
printf ("\n Procedencia Norte - Preco: %.2f \n", preco);
break;
case 3:
printf ("\n Procedencia Leste - Preco: %.2f \n", preco);
break;
case 4:
printf ("\n Procedencia Oeste - Preco: %.2f \n", preco);
break;
case 5:
case 6:
printf ("\n Procedencia Nordeste - Preco: %.2f \n", preco);
break;
case 7:
case 8:
case 9:
printf ("\n Procedencia Sudeste - Preco: %.2f \n", preco);
break;
case 10:
printf ("\n Procedencia Centro-Oeste - Preco: %.2f \n", preco);
break;
default:
printf ("\n Codigo nao encontrado! \n");
}

system ("PAUSE");
}
```

Área de execução do programa

Digite o codigo:
codigo = 6
Digite o preco:
preco: 2300

Simulação da execução do desvio condicional com múltipla escolha em C - Exemplo 8

Fonte: Elaborado pelo professor conteudista.

Exercícios Complementares

Escreva o programa em C para os seguintes itens:



1. Calcule e mostre a média aritmética tendo como base 4 avaliações (AV1, AV2, AV3 e AV4). Baseado na média, mostre a situação do aluno como aprovado se a media for maior ou igual a 6 e reprovado se for menor do que 6.
2. Verifique e mostre se um número inteiro é par ou ímpar.
3. Verifique e mostre se um número real é positivo ou negativo.
4. Dado o salário de um funcionário, faça o acréscimo de 10% caso o salário seja menor ou igual a 800; caso contrário faça o acréscimo de 20%. Por fim, mostre o novo salário.
5. Dado um número inteiro x, adicione 5 ao seu valor caso ele seja igual a zero. Se ele for menor do que zero, multiplique-o por 3; caso contrário, subtraia 7 ao seu valor. Por fim, mostre o novo valor de x.
6. Considerando o peso e altura de uma pessoa, calcule o IMC e mostre a situação da pessoa conforme a tabela abaixo:

| IMC | Situação |
|-------------------|----------------|
| Abaixo de 18.5 | Abaixo do peso |
| Entre 18.6 e 24.9 | Peso ideal |
| Acima de 24.9 | Acima do peso |

OBS: Para calcular o IMC, utilize a seguinte fórmula: $IMC = \text{peso}/(\text{altura}*\text{altura})$.

Quiz

Exercício Final

Implementação de desvios condicionais em C

INICIAR >



Referências

FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

RISSETTI, G.; PUGA, S. Lógica de Programação e estruturas de dados, com aplicações em Java. São Paulo: Prentice Hall, 2003.

SOUZA, M. A. F. e outros. Algoritmos e lógica de programação: um texto introdutório para Engenharia (2a. edição). São Paulo: Cengage Learning, 2011.



Avalie este tópico



ANTERIOR

<

Desvio condicional com múltipla escolha

Índice

≡

Biblioteca

(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Ajuda?

PRÓXIMO

(<https://ava.uninove.br/estruturas-de-repeticao-engenharia>)

Estruturas de repetição: engenharia

>

© Todos os direitos reservados