



Introdução a Programação Estruturada e Algoritmos

com implementação em linguagem C

AULA 3

mferoc

Tópicos

Operadores de Atribuição Aritmética e Operadores Incrementais

- Forma reduzida

- Operadores incrementais

Laços de Repetição

- O laço FOR

- O laço WHILE

- O laço DO...WHILE

- Loop infinito

Referências

Forma reduzida

Podemos utilizar uma forma reduzida para fazer atribuições aritméticas, facilitando ao fazer declarações longas e melhorando o entendimento do código. Considere uma variável do tipo inteiro chamada x e inicializada com o valor 5 (int x = 5), então teríamos as seguintes abreviações:

Exemplo na forma normal	Exemplo na forma reduzida
x = x + 2	x += 2
x = x - 2	x -= 2
x = x * 2	x *= 2
x = x / 2	x /= 2
x = x % 2	x %= 2

Tabela: Operadores de atribuição aritméticos na forma reduzida

Operadores incrementais

Para trabalhar com laços de repetição usaremos bastante o incremento de uma unidade (somar 1). Sabemos fazer isso das seguintes formas:

```
x = x + 1;  
x += 1;
```

Porém podemos simplificar ainda mais usando a seguinte sintaxe:

Forma normal	Operador de incremento
y = x; x = x + 1;	y = x++;
y = x; x = x - 1;	y = x--;
x = x + 1; y = x;	y = ++x;
x = x - 1; y = x;	y = --x;

Tabela: Operadores de incremento

Operadores incrementais

Exemplo de utilização do pré incremento e pós incremento.

```
#include <stdio.h>

int main() {

    int x = 5, y = 5;

    printf("%d\n", x);
    printf("%d\n", x++);
    printf("%d\n", x);
    printf("%d\n", y);
    printf("%d\n", ++y);
    printf("%d\n", y);

    return 0;
}
```

Laços de Repetição

Os laços de repetição (loops) são usados na programação para repetir um bloco de código até que uma condição específica seja atendida. Temos três tipos de laços em C, que são:

- for
- while
- do... while

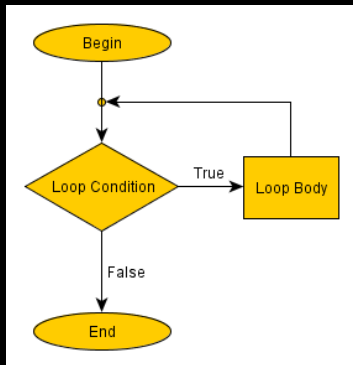


Figura: Estrutura básica de um laço de repetição.

O laço FOR

O laço FOR tem a seguinte estrutura:

```
#include <stdio.h>

int main() {

    for (initializationStatement; testExpression; updateStatement)
    {
        // codes
    }

    return 0;
}
```

O laço FOR

A instrução de inicialização (contador) é executada apenas uma vez. Então, a expressão de teste (condição) é avaliada. Se a expressão de teste for falsa (0), o laço será finalizado. Mas se a expressão de teste for verdadeira (diferente de zero), os códigos dentro do corpo do FOR serão executados e a expressão de atualização será atualizada (incremento no contador). Esse processo é repetido até que a expressão de teste seja falsa.

O laço FOR é comumente usado quando o número de iterações é conhecido. Para saber mais sobre a expressão de teste (quando a expressão de teste é avaliada como diferente de zero (verdadeiro) e 0 (falso)), verifique os operadores relacionais e lógicos.

O laço FOR

Veja no diagrama como é executado o laço FOR:

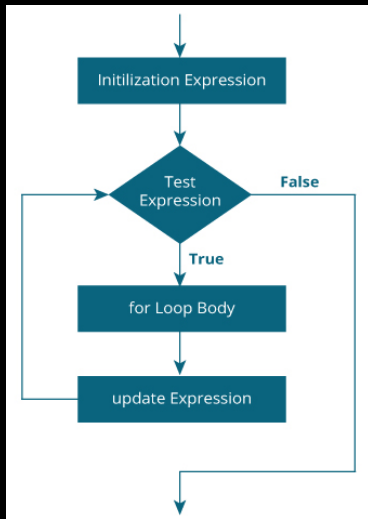


Figura: Diagrama do fluxo do laço FOR.

O laço FOR

Exemplo:

```
//Programa para calcular a soma dos primeiros n numeros naturais.  
//Numeros inteiros positivos 1,2,3 ... n sao conhecidos como  
//numeros naturais.  
  
#include <stdio.h>  
  
int main() {  
  
    int numero, contador, soma = 0;  
  
    printf("Entre um numero positivo e inteiro: ");  
    scanf("%d", &numero);  
  
    // o loop for termina quando contador eh menor ou igual a n  
    for(contador = 1; contador <= numero; contador++) {  
        soma += contador;  
    }  
  
    printf("Soma = %d", soma);  
  
    return 0;  
}
```

O laço FOR

Exercícios sobre FOR:

Ex. 1

Faça um programa em C com 2 laços de repetição FOR. O primeiro irá imprimir todos os números de 0 a 10, e o segundo irá imprimir todos os números de 10 a 0.

Ex. 2

Faça um programa em C que solicite ao usuário um número n , e mostre todos os números primos de 1 até n , utilizando o laço FOR.

O laço WHILE

O laço WHILE tem a seguinte estrutura:

```
#include <stdio.h>

int main() {

    while (testExpression)
    {
        //codes
    }

    return 0;
}
```

O laço WHILE

O laço WHILE avalia a expressão do teste. Se a expressão de teste for verdadeira (diferente de zero), os códigos dentro do corpo do laço WHILE serão executados. A expressão de teste é avaliada novamente. O processo continua até que a expressão de teste seja falsa. Quando a expressão de teste é falsa, o laço WHILE é finalizado.

O laço WHILE

Veja no diagrama como é executado o laço WHILE:

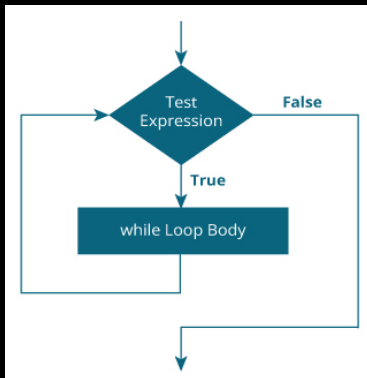


Figura: Diagrama do fluxo do laço WHILE.

O laço WHILE

Exemplo:

```
//Programa para calcular a soma dos primeiros n numeros naturais.  
//Numeros inteiros positivos 1,2,3 ... n sao conhecidos como  
//numeros naturais.  
  
#include <stdio.h>  
  
int main() {  
  
    int numero, contador, soma = 0;  
  
    printf("Entre um numero positivo e inteiro: ");  
    scanf("%d", &numero);  
  
    contador = 1;  
  
    // o loop while termina quando contador eh menor ou igual a n  
  
    while(contador <= numero) {  
        soma += contador;  
        contador++;  
    }  
  
    printf("Soma = %d", soma);  
  
    return 0;  
}
```

O laço WHILE

Exercícios sobre WHILE:

Ex. 1

Faça um programa em C com 2 laços de repetição WHILE. O primeiro irá imprimir todos os números de 0 a 10, e o segundo irá imprimir todos os números de 10 a 0.

Ex. 2

Faça um programa em C que solicite ao usuário um número n, e mostre todos os números primos de 1 até n, utilizando o laço WHILE.

O laço DO...WHILE

O laço DO... WHILE tem a seguinte estrutura:

```
#include <stdio.h>

int main() {

    do
    {
        // codes
    }
    while (testExpression);

    return 0;
}
```

O laço DO...WHILE

O bloco de código dentro das chaves é executado uma vez. Então, a expressão de teste é avaliada. Se a expressão de teste for verdadeira, o bloco dentro do laço será executado novamente. Esse processo continua até que a expressão de teste seja avaliada como 0 (falsa). Quando a expressão de teste é falsa (diferente de zero), o laço do ... while é finalizado.

O laço DO...WHILE

Veja no diagrama como é executado o laço DO...WHILE:

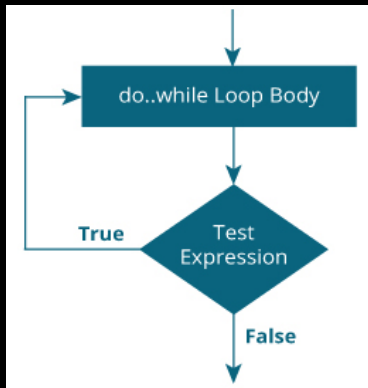


Figura: Diagrama do fluxo do laço DO...WHILE.

O laço DO...WHILE

Exemplo:

```
//Programa para calcular a soma dos primeiros n numeros naturais.  
//Numeros inteiros positivos 1,2,3 ... n sao conhecidos como  
//numeros naturais.  
  
#include <stdio.h>  
  
int main() {  
  
    int numero, contador, soma = 0;  
  
    printf("Entre um numero positivo e inteiro: ");  
    scanf("%d", &numero);  
  
    contador = 1;  
  
    // o loop do...while termina quando contador eh //menor ou igual  
    a n  
  
    do {  
        soma += contador;  
        contador++;  
  
    } while(contador <= numero);  
  
    printf("Soma = %d", soma);  
  
    return 0;  
}
```

O laço DO...WHILE

Exercícios sobre DO...WHILE:

Ex. 1

Faça um programa em C com 2 laços de repetição DO...WHILE. O primeiro irá imprimir todos os números de 0 a 10, e o segundo irá imprimir todos os números de 10 a 0.

Ex. 2

Faça um programa em C que solicite ao usuário um número n, e mostre todos os números primos de 1 até n, utilizando o laço DO...WHILE.

Loop infinito

Um loop que se repete indefinidamente e nunca termina é chamado de loop infinito. Na maioria das vezes, criamos loops infinitos por engano. No entanto, isso não significa que os loops infinitos não sejam úteis. Loops infinitos são comumente usados em programas que permanecem em execução por longos períodos até serem parados.

Loop infinito

Exemplo:

```
//Loop infinito com for.  
  
#include <stdio.h>  
  
int main() {  
  
    for(;;) {  
  
        printf("10");  
    }  
  
    return 0;  
}
```

Loop infinito

Exemplo:

```
//Loop infinito com while.  
  
#include <stdio.h>  
  
int main() {  
  
    while(1) {  
        printf("10");  
    }  
  
    return 0;  
}
```


Referências



Site Programiz

Programiz

<https://www.programiz.com/c-programming/c-for-loop>

<https://www.programiz.com/c-programming/c-do-while-loops>

<https://www.programiz.com/c-programming/c-do-while-loops>



prof. Salvador Alves de Melo Júnior (2019)

Material da disciplina Algoritmos e Lógica de Programação, aulas 3 e 4, da Faculdade Projeção de Sobradinho

