

Cláudio C. Rodrigues
Faculdade da Computação - UFU

° **LINGUAGEM C – 03: ESTRUTURAS DE REPETIÇÃO**

1

Objetivo

- Apresentar as estruturas de repetição **while**, **do...while** e **for...** as quais nos permitem executar um trecho de algoritmo **zero** ou **mais vezes**.

Apresentar os *comandos de controle* associados com as estruturas de repetição.

Apresentar os *comandos de desvio de fluxo incondicional*.

Estruturas de Repetição

1. Estrutura de repetição para número definido de repetições
 - Estrutura **for**
2. Estrutura de repetição para número indefinido de repetições
 - ✓ Repetição com pré-condição
 - Estrutura **while**
 - ✓ Repetição com pós-condição
 - Estrutura **do... while**
3. Comandos de *desvio incondicional*.
 - **break**, **continue**, **exit** e **goto**

Terminologia

- O código repetido em um laço (*loop*) é chamado **corpo do laço**.
 - ✓ O corpo do laço é constituído de um ou mais comandos.
- Cada repetição do *corpo do laço* é chamada de uma **iteração** do laço.
 - ✓ A *iteração* é controlada por um *teste de condição*.
- A **condição de controle** é uma expressão *booleana*.

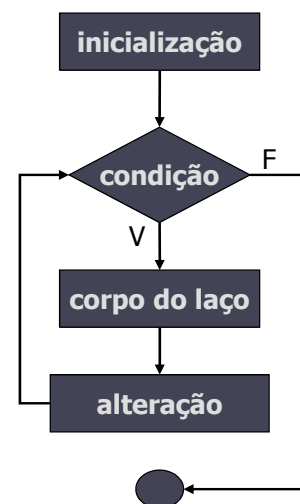
Estrutura de Repetição para Número Definido de Repetições

- tem seu funcionamento controlado por uma variável que conta o número de vezes que o comando é executado.
- Em **C**, essa estrutura é implementada pelo comando **for**, cuja forma básica é a seguinte:

```
for( inicialização; condição; alteração) {  
    comandos;  
}
```

```
for ( inicialização; condição; alteração) {  
    corpo do laço;  
}
```

- A **inicialização** é uma expressão que atribui um valor inicial ao contador
- A **condição** verifica se a contagem chegou ao fim
- A **alteração** modifica o valor do contador.
- Enquanto a contagem não termina, o *corpo do laço* associado ao **for** é repetidamente executado.



Programa exemplo 1:

```
#include <stdio.h>
int main()
{
    int up, down;
    for (up = 0, down=10; up < down; up++, down--)
    {
        printf("up = %d, down= %d\n",up,down);
    }
    return 0;
}
```

Programa exemplo 2:

- Exibir a tabela ASCII com pausas a cada 23 linhas.

```
#include <stdio.h>
int main() {
    int c, n=0;
    for(c=0; c<=255; c++) {
        printf("\n%c ==> %d", c, c);
        n++;
        if( n==23 ) {
            printf("\n\nPressione uma tecla ...");
            n=0;
            getch();
        }
    }
    return 0;
}
```

Desafios de Programação

1. O quadrado de um número natural **n** é dado pela soma dos **n** primeiros números ímpares consecutivos. Por exemplo, $1^2=1$, $2^2=1+3$, $3^2=1+3+5$, $4^2=1+3+5+7$, etc.
 - Dado um número **n**, calcule seu quadrado usando a soma de ímpares ao invés de produto.

Estrutura de Repetição para Número Indefinido de Repetições

- Essa estrutura de repetição é utilizada quando **não** se sabe o número de vezes em que um trecho do algoritmo deve ser repetido.
 - ✓ Embora, também possa ser utilizada quando se sabe esse número.
- **C**, implementa dois comandos para esta situação:
 - Comando **while**
 - Comando **do... while**

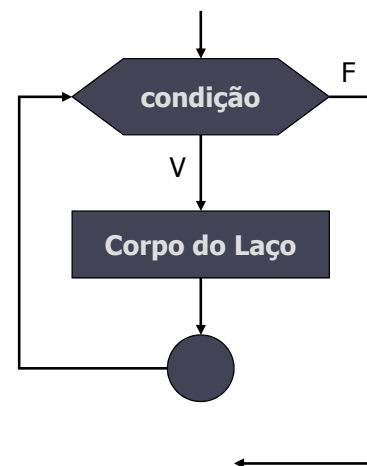
Estrutura de Repetição para Número Indefinido de Repetições

- A diferença importante entre os laços **while** e **do-while** envolve o *momento* em que a *expressão booleana* de controle é verificada.
- Com **while**,
 - ✓ a *expressão booleana* é verificada **antes** que o corpo do laço seja executado.
- Com **do-while**,
 - ✓ o *corpo do laço* é executado *primeiro*, e a *condição* verificada **depois**.

Repetição com pré-condição

- **while**
 - ✓ O corpo do laço é executado **zero** ou **mais vezes**
- Sintaxe

```
while( condição )  
{  
    corpo do laço;  
}
```



Programa exemplo 3:

// Exibir os dígitos de um número.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    unsigned int n, d;
    printf("\n Digite um número: ");
    scanf("%u", &n);
    printf("\n Os seus dígitos são: ");
    while( n != 0 ) {
        d = n % 10;
        n /= 10;
        printf("%u ", d);
    }
    getch();
    return 0;
}
```

Desafio de Programação

1. uma certa agência bancária, as contas são identificadas por números de até seis dígitos seguidos de um dígito verificador, calculado conforme exemplificado abaixo. Dado um número de conta **n**, exiba o número de conta completo correspondente.
 - Seja **n** = 7314 o número da conta.
 - Adicionamos os dígitos de **n** e obtemos a soma **s** = 4+1+3+7 = 15;
 - Calculamos o resto da divisão de **s** por 10 e obtemos o dígito **d** = 5.
 - Número de conta completo: **007314-5**

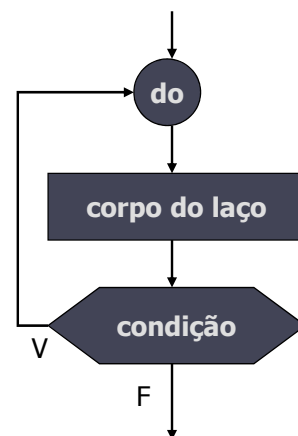
Desafio de Programação

1. Um número natural é *triangular* se é igual à soma dos **n** primeiros números naturais consecutivos, a partir de 1.
 - Por exemplo, 1, 3, 6, 10, 15, ... são *triangulares*.
- Dado um natural **n** ≥ 1 , informe se ele é *triangular*.

Repetição com pós-condição

- **do-while**
 - ✓ O corpo do laço é executado **uma** ou **mais vezes**
- Sintaxe

```
do
{
    corpo do laço;
} while( condição);
```



Programa exemplo 4:

```
#include <stdlib.h>
int main(){
    int x;

    do{
        printf("Digite um inteiro positivo: ");
        scanf("%d",&x);
    }while(x<=0);
    printf("O valor digitado foi %d\n",x);
    return 0;
}
```

Comandos de desvio incondicional

- As vezes é preciso alterar o fluxo de controle de uma estrutura de laço.
- As duas formas para se alterar o fluxo de controle são:
 - comando **break** ou **continue**.
- **break**
 - Encerra o laço.
- **continue**
 - Encerra a iteração atual do corpo do laço.

Programas exemplos 5:

break & continue

```
#include <stdio.h>
int main() {
    int n, k;
    printf("\nDigite um número
    natural: ");
    scanf("%u", &n);
    for(k=2; k<=n-1; k++)
        if( n%k == 0 ) break;
    if( k==n )
        printf("\nO número é primo");
    else printf("\nO número não é
    primo");
    getch();
    return 0;
}
```

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

int main() {
    int i=0;
    printf("Os cinco primeiros
    numeros pares:\n");
    for(i=0; i<10; i++)
    {
        if(i%2) continue;
        printf("%d ", i);
    }
    printf("\n");
    getch();
    return 0;
}
```

Laços Aninhados

- É perfeitamente legal aninhar um laço dentro do outro.
- Quando fizer isto, lembre-se de que qualquer comando **break** ou **continue** se aplicam ao laço mais interno que o contenha.
- Exemplo:

```
for(i=0; i<10; i++)
{
    for(j=0; j<10; j++)
        printf("%d ", j);
    printf("\n");
}
```



Bibliografia

- Pereira, S.L., Linguagem C – Distribuição gratuita
- Schildt, H., C Completo e Total, Editora Makron Books do Brasil Editora Ltda, 1996.
- Evaristo, J., Aprendendo a programar programando em linguagem C, Book Express, 2001.
- Mizrahi, V. V., Treinamento em Linguagem C, Curso Completo, Módulos 1 e 2, Makron Books do Brasil Editora Ltda, 1990.
- Kernighan, B.W & Ritchie, D. M., C a Linguagem de Programação, Editora Campus, 1986.



Dúvidas?