

Linguagem de Programação I - ESP201

Prof^a Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal de Ciência e Tecnologia da Bahia
Campus de Feira de Santana

carolsoko@ifba.edu.br

December 3, 2024

Estruturas de Repetição

1 Estruturas de Repetição

■ Estrutura de Repetição FOR

- Estrutura de Repetição FOR - Vantagens
- Estrutura de Repetição FOR - LOOP INFINITO
- Estrutura de Repetição FOR - LOOP VAZIO
- Estrutura de Repetição FOR - Exercícios

2 Referências

Estruturas de Repetição

Estruturas de Repetição

Suponha que você deseja fazer o seguinte código:

```
#include <stdio.h>
voidmain(){
    printf("1 ");
    printf("2 ");
    :::
    printf("100");
}
saída: 1 2 3 4 ... 100
```

Estruturas de Repetição

As Estruturas de Repetição permitem que um bloco (ou lista) de comandos seja executado repetidamente, até que uma determinada condição de interrupção seja satisfeita. A condição de interrupção é representada por uma expressão lógica [Schildt e Mayer 1997].

Na linguagem C, existem três tipos de Estruturas de Repetição:

- 1 for
- 2 while
- 3 do-while (em outras linguagens conhecido como repeat until)

Essas estruturas de repetição são fundamentais para repetir comandos, ou trechos de códigos, sucessivas vezes.

Estruturas de Repetição FOR

Estrutura de Repetição FOR

Uma estrutura de repetição examina uma ou mais condições e executa um bloco de instruções sempre que a condição for atendida.

A Estrutura de Repetição FOR é uma estrutura de Repetição muito utilizada em todas as linguagens de programação.

O FOR executa um número determinado de vezes, ele define um valor inicial e um valor final e o INCREMENTO ou DECREMENTO, dependendo da necessidade do algoritmo.

```
for (<início>; <condição>; <incremento>){  
    instrução ou instruções que serão repetidas;  
}
```

Estrutura de Repetição FOR

Vejamos o exemplo anterior, como fica resolvido com FOR:

```
1  #include <stdio.h>
2  void main(){
3      for(int num = 1; num <= 100; num++)
4          printf("%d ", num);
5  }
6
```

input

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
92 93 94 95 96 97 98 99 100
```


Estrutura de Repetição FOR

```
for (<início>; <condição>; <incremento>){  
    instrução ou instruções que serão repetidas;  
}
```

Na forma mais simples:

- **Inicialização:** expressão de atribuição sempre executada uma única vez.
- **Teste:** condição que controla a execução do laço é sempre avaliada a cada execução.
 - *verdadeiro* → continua a execução
 - *falso* → para a execução
- **Incremento ou Decremento:** define como a variável de controle será alterada Este incremento é sempre executado após a execução do corpo do laço.

Estrutura de Repetição FOR - FLEXIBILIDADE

- **FLEXIBILIDADE:** qualquer expressão de um laço "for" pode conter várias instruções separadas por vírgula, tanto de inicialização, quanto de teste ou incremento.

```
1  #include <stdio.h>
2  void main(){
3      for (int x=0, y=0; (x+y)<100; ++x, y=y+x)
4          printf("%d ", x+y);
5
6  }
```

0 2 5 9 14 20 27 35 44 54 65 77 90

Estrutura de Repetição FOR - Contagem Simples

```
1  #include <stdio.h>
2  void main(){
3      int i,j,resposta;
4      char feito = 'S';
5      for (i=1;i<100 && feito != 'N';i++) {
6          for (j=1;j<10;j++) {
7              printf("Quanto é %d + %d? ",i,j);
8              scanf("%d",&resposta);
9              getchar(); //limpar o buffer por causa do <enter>
10             if (resposta != (i+j))
11                 printf("Errou!\n");
12             else
13                 printf("Acertou!\n");
14         }
15         printf("Mais? (S/N) ");
16         scanf("%c",&feito);
17         getchar(); //limpar o buffer por causa do <enter>
18     }
19 }
```

Estrutura de Repetição FOR - LOOP INFINITO

- **LOOP INFINITO:** podemos fazer um loop infinito usando a estrutura de repetição FOR de forma bem simples.
- A ausência de condições de inicialização, incremento e terminação causam um processo contínuo e "teoricamente" infinito.

```
for ( ; ; ){  
    instrução ou instruções para repetição;  
}
```

Estrutura de Repetição FOR - BREAK

- **BREAK:** Ao contrário de outras linguagens que não permitem o término do **loop** a não ser quando a condição de finalização for satisfeita, a linguagem C permite que um **loop** seja interrompido antes de seu término sem que exista qualquer tipo de inconveniente. O comando "**break**" causa essa interrupção.

Usando BREAK com LOOP INFINITO

```
for ( ; ; ){  
    instrução ou instruções para repetição;  
    if(condição do break) break;  
}
```

Estrutura de Repetição FOR - BREAK

- **BREAK:** Ao contrário de outras linguagens que não permitem o término do **loop** a não ser quando a condição de finalização for satisfeita, a linguagem C permite que um **loop** seja interrompido antes de seu término sem que exista qualquer tipo de inconveniente. O comando "**break**" causa essa interrupção.

Usando BREAK com um FOR qualquer

```
for (<início>; <condição>; <incremento>){  
    instrução ou instruções para repetição;  
    if(condição do break) break;  
}
```

Estrutura de Repetição FOR - BREAK

Vejamos um exemplo prático do uso do **BREAK**:

```
1  #include <stdio.h>
2  void main(){
3      char c;
4      for(;;) {
5          printf("Digite uma letra do alfabeto\n");
6          printf("Se você adivinhar a letra, poderá acessar o sistema: ");
7          scanf("%c",&c); getchar();
8          if (c == 'A')
9              break;
10         /*interrompe o que deveria ser um laço infinito*/
11     }
12     printf("Acesso permitido!");
13 }
```

Estrutura de Repetição FOR - LOOP VAZIO

- **LOOP VAZIO:** podemos fazer um loop vazio usando a estrutura de repetição FOR de forma simples, com o intuito apenas de realizar a repetição.
- A presença do ponto e vírgula finalizando o comando, força a execução do **loop** sem que seja executado qualquer comando.

```
for (<início>; <condição>; <incremento>);
```


Estrutura de Repetição FOR - Exercícios

- 1 Faça um algoritmo que leia 20 valores e calcule a média deles.
- 2 Faça um algoritmo que implemente um sistema de LOGIN e SENHA. **LOGIN = "ifbaBSI"** e **SENHA = "alunoIFBA"**. O sistema deverá informar "Acesso Negado", caso o usuário erre login e/ou senha. E deverá informar "Acesso Permitido", caso contrário. O algoritmo deverá permitir que o usuário erre apenas 5 vezes. Após 5 erros, o usuário terá a senha bloqueada e não poderá tentar logar novamente. Informe, a cada erro, quantas tentativas o usuário ainda tem.
- 3 Faça um algoritmo que leia 50 valores, encontre e mostre o maior e o menor deles.
- 4 Faça um algoritmo que leia um número inteiro de 0 a 9 e informe a tabuada da multiplicação, soma e subtração desse número.

Referências

Referências



SCHILDT, H.; MAYER, R. *C completo e total*. [S.l.]: Pearson University, 1997. ISBN 9788534605953.