

# Técnicas de Programação I

Estruturas de Repetição

Prof. José Luiz

## Comando for

A instrução **for** (ou laço **for**), adapta-se particularmente a situações em que o número de iterações é conhecido *a priori*. A sua sintaxe é:

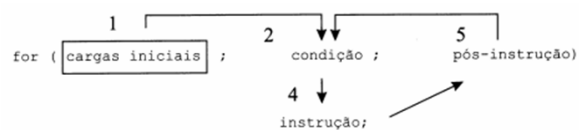
**for (cargas iniciais; condição; pós-instrução)**

```
{  
    instrução;  
    instrução;  
}
```

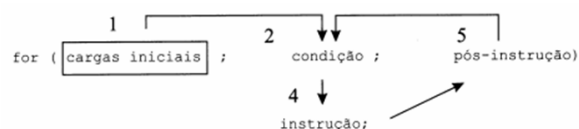
## Comando for

**for (cargas iniciais; condição; pós-instrução)  
instrução;**

O seu funcionamento pode ser resumido pelo esquema:



## Comando for



1. Aqui são iniciadas as variáveis presentes no laço. Ele é executado apenas uma vez.
2. A condição é avaliada.
3. Se condição FALSA, então o laço **for** termina.
4. Se condição VERDADEIRA, então é executada a **instrução** ou bloco de instruções do laço.
5. Depois de executada a instrução do laço é executada a **pós-instrução** (tipo incremento e decremento)
6. Voltar ao ponto 2

Escreva um programa que calcule a soma e o produto dos  $n$  primeiros números naturais

```
int main()
{
    int n, num, soma, produto;
    printf("Digite um numero: ");
    scanf("%d",&num);
    for(soma=0, n=1,produto=1; n<=num; n++)
    {
        soma = soma + n;
        produto = produto * n;
    }
    printf("Soma = %d\nProduto=%d\n", soma, produto);
    system("pause");
}
```

*Nota:*

O laço for identifica os seus três componentes, separando-os por ponto-e-vírgula (;). Assim, se for necessário realizar mais do que uma carga inicial ou mais do que uma pós-instrução, estas deverão ser separadas por vírgula (,).

Escreva um programa que coloque na tela as 5 primeiras tabuadas, parando a tela depois de cada uma delas ser escrita.

```
int main()
{
    int i, j;
    for (i=1 ; i<=5 ; i++)
    {
        for (j=1 ; j<=10 ; j++)
            printf("%d * %d = %d\n", i ,j, i*j);
        if (i != 5)
            system("pause");
    }
}
```

Escreva um programa que coloque na tela os 10 primeiros números inteiros.

```
int main()
{
    int i;
    for (i=1 ; i<=10 ; i++)
    {
        printf("%d\n", i);
    }
    system("pause");
}
```

Escreva um programa que some os inteiros ímpares entre 1 e 99 usando a estrutura *for*.

```
int main()
{
    int i, soma=0;
    for (i=1 ; i<=99; i++)
    {
        if (i%2 != 0)
            soma = soma + i;
    }
    printf("%d\n", soma);
    system("pause");
}
```

Escreva um programa que apresente os seguintes números na tela:

```
1
1 2
1 2 3
.....
1 2 3 4 5 6 7 8 9 10
```

```
int main()
{
    int i, j;
    for (i=1 ; i<=10; i++)
    {
        for (j=1;j<=i;j++)
            printf("%d ", j);
        printf("\n");
    }
    system("pause");
}
```

Escreva um programa que solicite o usuário um número e que escreva simultaneamente a sequência crescente e decrescente entre 1 e esse número.

```
1      3
2      2
3      1
```

```
int main()
{
    int i, j, n;
    printf("Digite N: ");
    scanf("%d", &n);
    for (i=1, j=n ; i<=n; i++, j--)
    {
        printf("%d \t %d\n", i, j);
    }
    system("pause");
}
```

Escreva um programa que solicite o usuário um número. Em seguida escreva todos os números inteiros a partir desse número, exceto os múltiplos de 3. Quando encontrar o primeiro múltiplo de 10 a execução termina.

```
N = 13
13 14 16 17 19

int main()
{
    int i, n;
    printf("Digite N: ");
    scanf("%d",&n);
    for (i=n, ; ; i++) // laço infinito
    {
        if (i % 10 == 0) // múltiplo de 10
            break;
        else
            if (i % 3 == 0)
                continue;
        printf("%d \n", i);
    }
    system("pause");
}
```