

5ª Aula prática

Estrutura de repetição “for”

Seu formato é

```
for (inicialização; condição; incremento)  
    comando interno;
```

e sua finalidade é repetir o “*comando interno*” enquanto a “*condição*” permanecer verdadeira. É semelhante à estrutura “*while*” da aula anterior, mas a estrutura “*for*” prevê campos específicos para “*inicialização*” e para estabelecer “*incremento*”. É especialmente projetada para executar uma ação repetida sob controle de um contador que é incrementado em cada iteração (repetição).

Ela trabalha da seguinte forma:

1. *inicialização* é executada; visa, geralmente, a atribuição de valor a uma variável contadora;
2. *condição* é verificada; se é verdadeira a repetição continua, senão a repetição é encerrada e o “*comando interno*” não é mais executado;
3. *comando interno* é executado. Na verdade, pode ser um comando composto, ou bloco de comandos, entre chaves { };
4. finalmente, *incremento* é executado e o processo retorna para o passo 2 acima.

Exemplo 1 : programa para calcular o fatorial de um número.

```
#include <iostream>  
using namespace std;  
main()  
{  
    int x, fat;  
    cout << "Digite um número inteiro: ";  
    cin >> x;  
    fat = 1;  
    for ( int i=2; i<=x; i=i+1 )  
        fat = fat * i;  
    cout << "fatorial: " << fat << endl;  
    system("pause");  
}
```

Exemplo 2 : programa que imprime todos os divisores de um número

```
#include <iostream>  
using namespace std;  
main()  
{  
    int x;  
    cout << "Digite um número: ";  
    cin >> x;  
    for ( int i=x; i >= 1; i=i-1 )  
        if ( x%i == 0 )  
            cout << i << " é divisor de " << x << endl;  
    system("pause");  
}
```

Observações:

1. estamos designando como *incremento* algo que, na verdade, pode ser um decremento, como no

- exemplo 2 acima;
2. a instrução de incremento $i=i+1$ pode ser escrita como $i++$; também a instrução $i=i-1$ pode ser escrita como $i--$;
 3. a estrutura “for” é muito mais flexível do que apresentado aqui. Quem quiser conhecer todas as possibilidades deve consultar um livro ou um tutorial;

Exemplo 3 : programa que escreve uma mensagem com a seguinte estrutura:

```

sonho
sonho sonho
sonho sonho sonho
sonho sonho sonho sonho
...
#include <iostream>
using namespace std;
main()
{
    int linhas;
    cout << "Quantas linhas: ";
    cin >> linhas;
    for (int lin=1; lin<=linhas; lin++) {
        for (int col=1; col<=lin; col++)
            cout << "sonho ";
        cout << endl;
    }
    system("pause");
}

```

Exercícios:

3) Crie um programa para imprimir a tabela de conversão de graus Celsius para Fahrenheit e Kelvin. A tabela deve abranger as temperaturas de 0°C a 100° C.

4) O valor aproximado do número π pode ser calculado usando-se a série:

$$S = 1/1^3 - 1/3^3 + 1/5^3 - 1/7^3 + 1/9^3 - \dots$$

sendo $\pi = (S * 32)^{1/3}$.

Escreva um programa para ler um número inteiro n, calcular e imprimir uma aproximação para o valor de π usando os n primeiros termos da série.

a) Altere o programa do exemplo 3 para imprimir o dobro do número de linhas especificado, formando uma configuração como exemplificado abaixo para número de linhas igual a 4.

```

sonho
sonho sonho
sonho sonho sonho
sonho sonho sonho sonho
sonho sonho
sonho

```

b) Altere o programa do exemplo 3 para imprimir o número de linhas especificado, com a seguinte configuração:

```

        sonho
        sonho sonho sonho
        sonho sonho sonho sonho sonho
sonho sonho sonho sonho sonho sonho sonho
.....

```