

BCC 201 - Introdução à Programação  
Controle de Fluxo  
Comandos de repetição: *for*

Guillermo Cámara-Chávez  
UFOP

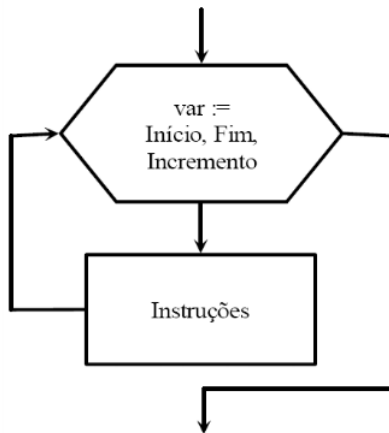
# Laços contados I

- ▶ São úteis quando **se conhece previamente o número exato de vezes** que se deseja executar um determinado conjunto de comandos
- ▶ **Estrutura** dotada de mecanismos **para contar o número de vezes** que o corpo do laço é executado.

## Laços contados II

```
for (início; condição de parada; incremento/decremento)
{
    conjunto de instruções
}
```

## Laços contados III



## Laços contados IV

Elaborar um programa para calcular e exibir as  $n$  parcelas da sequência:

2, 4, 6, 8, ...

## Laços contados V

```
int main()
{
    int i, num, n;
    cout << "Digite numero de parcelas \n";
    cin >> n;
    for (i = 1, num = 2; i <= n; i++, num+=2)
        cout << num;
    return 0;
}
```

## Laços contados VI

Escreva um programa em linguagem C++ para ler dois números inteiros e mostrar na tela o produto deles, sendo que este cálculo deve ser realizado através de somas sucessivas.

Exemplo:  $3 \times 5 = 3 + 3 + 3 + 3 + 3$  ou  $5 + 5 + 5$ .

## Laços contados VII

```
int main()
{
    int num1, num2, i, mult = 0;
    cout << "Inserir dois numeros";
    cin >> num1 >> num2;
    for (i = 1; i <= num2; i++)
        mult = mult + num1;
    cout << num1 << "x" << num2 << "=" << mult;
    return 0;
}
```



## Laços contados VIII

Escreva um programa para mostrar (em ordem decrescente) na tela todos os números pares entre um número fornecido via teclado e 0 (zero).

## Laços contados IX

```
int main()
{
    int num, i;
    cout << "Digite um numero \n";
    cin >> num;
    for (i = num; i >= 0; i--)
        if (i % 2 == 0)
            cout << i;
    return 0;
}
```

# Comandos de salto I

## O comando `break`

- ▶ Usando o comando `break`, podemos **sair** de um *loop* mesmo que a condição ainda seja verdadeira.
- ▶ Pode ser usado para **terminar** um *loop* infinito
- ▶ Por exemplo, podemos interromper a contagem regressiva antes do seu final

## Comandos de salto II

```
int main ()
{
    int n;
    for (n=10; n>0; n--) {
        cout << n << ", " ;
        if (n==3) {
            cout << "contagem regressiva cancelada!";
            break;
        }
    }
    return 0;
}
```

Saída na tela:

10, 9, 8, 7, 6, 5, 4, 3, contagem regressiva cancelada!!

# Comandos de salto III

## O comando `continue`

- ▶ Faz com que o programa ignore o restante dos comandos do bloco,
- ▶ é como se o fim do bloco tivesse sido alcançado,
- ▶ o programa salta para o início da próxima iteração.
- ▶ Por exemplo, saltaremos o número 5 na nossa contagem regressiva

## Comandos de salto IV

```
int main ()
{
    for (int n=10; n>0; n--) {
        if (n==5)
            continue;
        cout << n <<" ";
    }
    cout << "FOGO!\n";
    return 0;
}
```

Saída na tela:

10, 9, 8, 7, 6, 4, 3, 2, 1, FOGO!

## Exercício I

Fazer um programa em C que seja capaz de ler um número inteiro positivo do teclado e de verificar se a sua sequência de dígitos é exatamente a mesma, tanto se for analisado da esquerda para a direita quanto da direita para a esquerda.

Caso seja, imprimir “ESSE NÚMERO É PALÍNDROMO!!!”. Caso não seja, imprimir “ESSE NÚMERO NÃO É PALÍNDROMO.”

OBSERVAÇÃO: utilize divisões e/ou multiplicações sucessivas por 10.

## Exercício II

```
int main(){
    int num, numinv = 0, dig, numant;
    cout << "Digite um numero: ";
    cin >> num;
    numant = num;
    while(num > 0)
    {
        dig = num % 10;
        num = num / 10;
        numinv *= 10;
        numinv += dig;
    }
    if (numant == numinv)
        cout << "ESSE NÚMERO É PALÍNDROMO!";
    return 0;
}
```



## Exercício III

Escrever um programa para calcular a série abaixo e imprimir o resultado de  $x$ . O número de termos é definido pelo usuário.

$$x = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{n}$$

## Exercício IV

```
int main(){
    double x = 0;
    int i, termos;
    cout << "Inserir o numero de termos: ";
    cin >> termos;
    for (i = 1; i <= termos; i++)
    {
        if (i % 2 != 0) // impar
            s += 1.0/i;
        else
            s -= 1.0/i;
    }
    cout << "s = " << s;
    return 0;
}
```

## Exercício V

Crie um programa que leia um número entre 2 e 10 e gere uma tela com a seguinte configuração:

Digite um número :

4

Saida do programa :

```
****  
****  
****  
****
```

## Exercício VI

```
int main(){
    int i, j, num;
    cout << "Inserir numero: ";
    cin >> num;
    if (num > 1 && num < 11){
        for (i = 1; i <= num; i++){
            for (j = 1; j <= num; j++){
                cout << "*";
            }
            cout << "\n";
        }
    }
    return 0;
}
```

## Exercício VII

Dadas as coordenadas  $(x, y)$  de  $n$  pontos no Plano Cartesiano, elaborar um programa para verificar qual é o ponto mais próximo da origem dos eixos  $(0,0)$ . Mostrar a menor distância.

## Exercício VIII

```
int main(){
    float x, y, mindist, dist;
    int n, i;
    cout << "Digite o numero de pontos";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cout << "digite coordenadas";
        cin > x >> y;
        dist = sqrt(x*x+y*y);
        if (i == 0 || dist < mindist)
            mindist = dist;
    }
    cout << "A menor distancia eh: " << mindist;
    return 0;
}
```

FIM