

**9**

**Loop FOR (Para)**

# COMANDO Loop FOR (Para)

É um loop finito que repetirá um conjunto de instruções usando três configurações básicas.

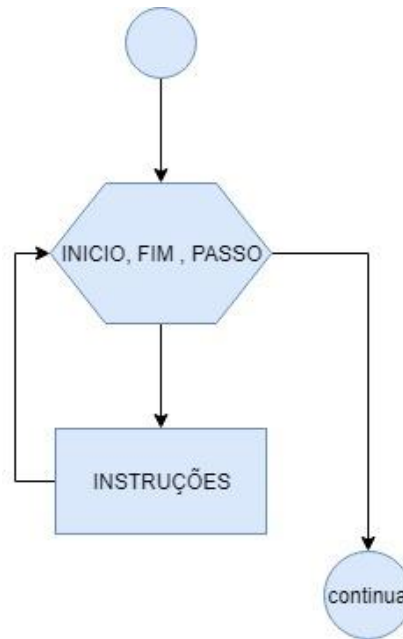
O símbolo usado denomina-se Processamento predefinido ou Preparação,

(Símbolo HEXAGONO)

**INÍCIO:** variável numérica que determina o início do loop, int i=0, int j = 1, int x = 2, etc;

**FIM:** Determinará o limite ou final do loop, i<10, j <> 20, x <= 2, etc;

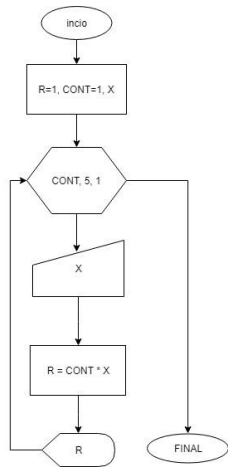
**PASSO OU INCREMENTO:** Determina o tipo de incremento ou passo, x++, x--, x=x+2, x=x+1, etc;



```
for ( INÍCIO; FIM; PASSO )  
{  
  ( INSTRUÇÕES QUE  
    SERÃO REPETIDAS )  
}
```

# COMANDO PARA ( FOR ...FAÇA )

No programa1 **cont** vai variar de 1 até 5 sempre de 1 em 1; No programa2 **Y** vai variar de 0 até 10 sempre de 2 em 2. 1) **Faça a simulação/teste no caderno;** 2) **Monte o programa completando o código que falta e indentando;** 3) **Execute os programas dentro do main()**

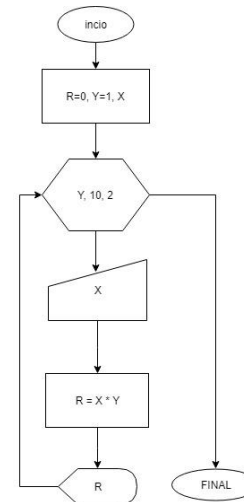


## PROGRAMA 1

```
int R=1, X;  
for (int cont = 1; cont <= 5; cont ++)  
{ cout << "Valor de X: "; cin >> X;  
  R = X * cont;  
  cout << cont << "º vez: valor de  
R=" << R << endl; }
```

## ATENÇÃO:

Fazer a simulação dos programas 1 e 2, lendo o X com os valores: 6 , 3, -1, 3 e 4.



## PROGRAMA 2

```
int R=1, X, Y = 1;  
for (Y=1; Y <= 10; Y=Y+2)  
{ cout << "Valor de X:";  
  cin >> X;  
  R = X * Y;  
  cout << "\nValor R= " << R << endl;  
}
```

# FOR dentro de um VOID

## PROGRAMA 3

Faça um programa que exiba os números não divisíveis por 3 entre 1 e 7.

*1) Faça a simulação/teste do void mostrar ( ) no caderno para os valores de x = 1 e y = 7, os valores para Z no caderno deverão ser iguais aos valores de Z impressos na tela;*

*2) Monte o programa completando com as bibliotecas o código que falta e indentando;*

*3) Apenas execute o programa do jeito que está para comparar os resultados com a simulação.*

```
void mostrar (int x, int y )
{
    int z; // valores não divisíveis por 3

    for (int i =x; i<=y; i++)
    { int resto = i % 3;
      if ( resto != 0 )
      { z= i;
        cout << "Z=" << z << endl;
      }
    }
    system("pause");
}

int main ( )
{
    mostrar ( 1 , 7 );
}
```

## TAREFA DE FIXAÇÃO

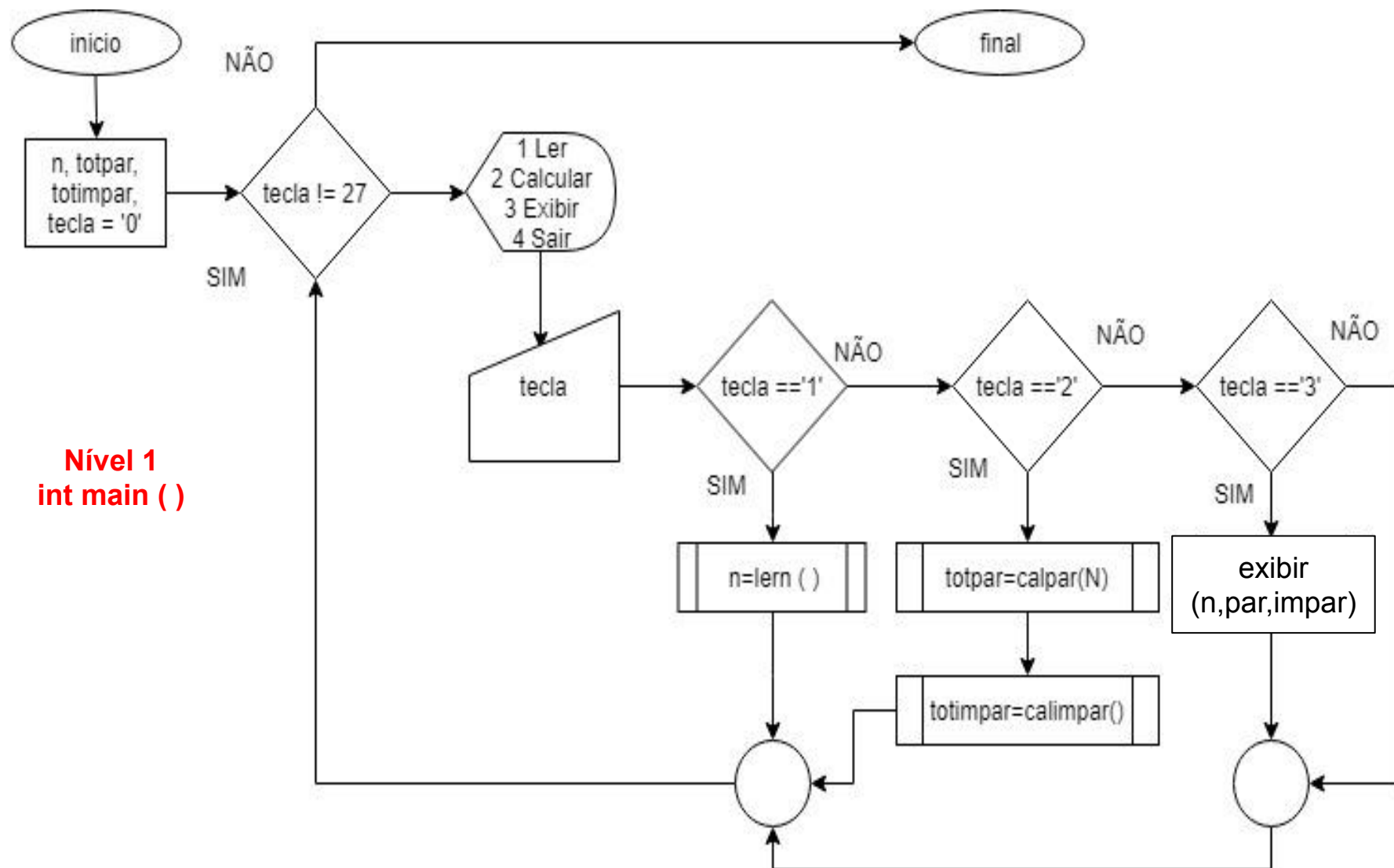
- a) O PROGRAMA 4 trata-se de um algoritmo contendo uma function para armazenar o somatório de todos os valores entre **n1** e **n2**, que são divisíveis por 3 e por 4 ao mesmo tempo e um VOID para mostrar o valor na tela. *Para este programa, monte e execute o código fonte. completando com as bibliotecas ou funções que faltarem, depois faça a simulação/teste da function soma (x, y) dentro do próprio código.*
  
- b) O PROGRAMA 5 trata-se de um Diagrama de Blocos Nível 1 e Nível 2. Para este programa, faça o código fonte baseado no diagrama, mas não se esqueça de acrescentar o void exibir (n, par, ímpar), que está faltando no nível 2, para exibir os totais de pares e de ímpares.

NOTA: Entregar os dois codigos em um arquivo ms-word juntamente com os prints de execução.

## PROGRAMA 4: Código Fonte

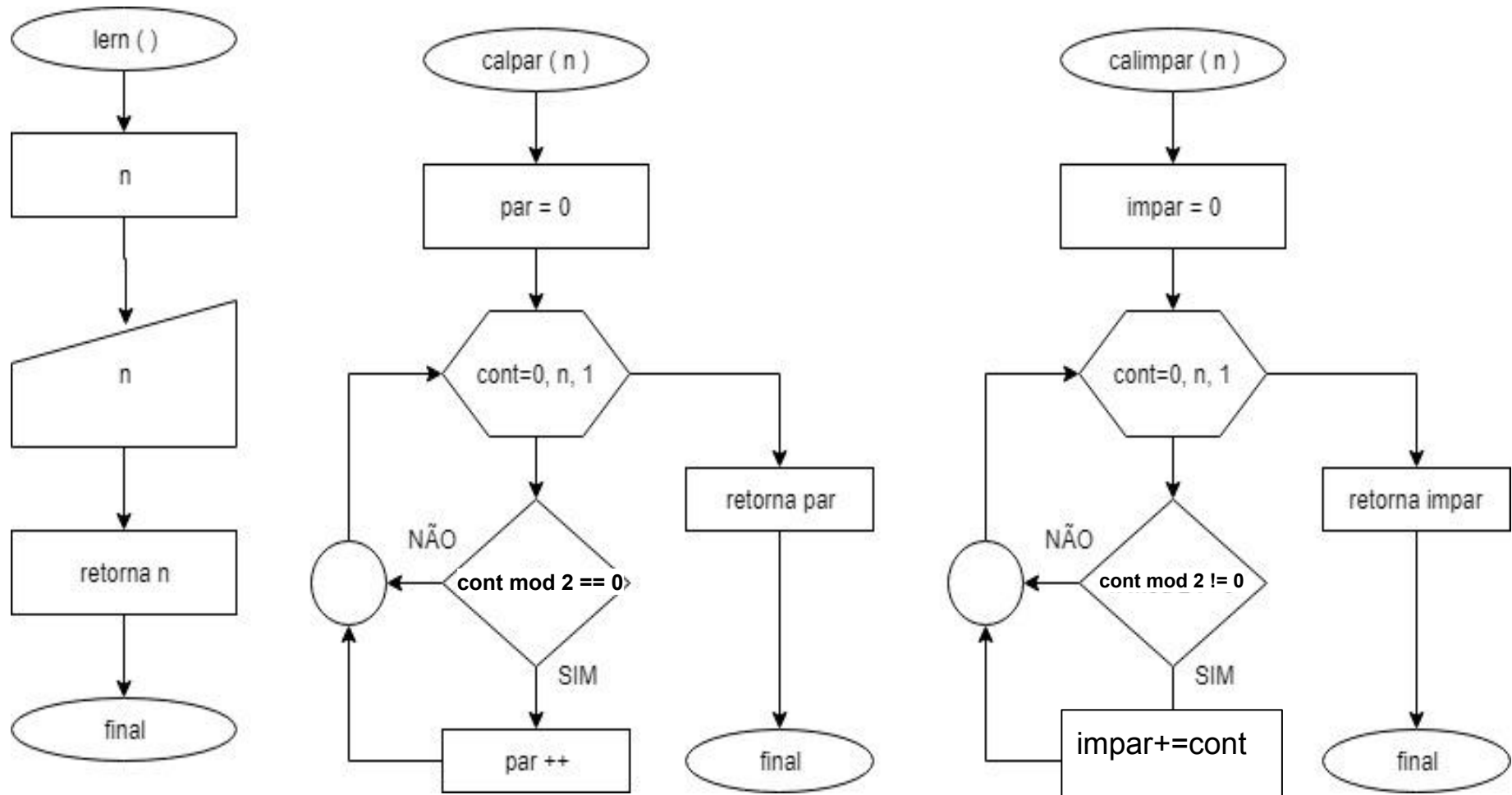
```
int soma (int x, int y ) {  
    int R=0, resto1, resto2;  
    for (int i =x; i<=y; i++) {  
        resto1 = i % 3;  resto2 = i % 4;  
        if ( resto1 == 0 && resto2 == 0 )  
            { R += i;  
              cout << "R="<<R<< endl; }  
    }  
    return R;    }  
  
void mostrar (int n1, int n2) {  
    int result = soma(n1,n2);  
    cout<< "Somatório dos divisíveis por 3 e 4 de "<< n1<< " até "<< n2<< "=";  
    cout << result << endl;  
    system("pause"); }  
  
int main ()  
{ setlocale(LC_ALL, "Portuguese" ); int n1=1, n2 = 13; mostrar ( n1 , n2 ); }
```

## PROGRAMA 5: Nível 1 - int main()



**Nível 1**  
**int main ( )**

## PROGRAMA 5: Nível 2 - SubRotinas



ATENÇÃO: Lembre-se que você deverá fazer o código do void exibir (n, par, impar) que está faltando.