



**USP - ICMC - SSC
SSC 0301 - 2o. Semestre 2013**

**Disciplina de
Introdução à Computação
para Engenharia Ambiental**

Prof. Dr. Fernando Santos Osório

LRM - Laboratório de Robótica Móvel do ICMC / CROB-SC

Email: fosorio@icmc.usp.br ou fosorio@gmail.com

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Material on-line:

Wiki ICMC - <http://wiki.icmc.usp.br/index.php>

Wiki SSC0301 - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))

Linguagem de Programação “C”

Agenda:

- **Programas Sequenciais x Programas com Repetição**
- **Comandos de Repetição (laços): FOR e WHILE**
 - > **Fluxo de Execução com Laços FOR**
 - > **Fluxo de Execução com Laços WHILE**
 - > **Comandos FOR e WHILE: Contadores**
 - > **Comandos FOR e WHILE: Acumuladores (Somadores)**
 - > **Comandos FOR e WHILE: Aninhados**
 - > **Exercícios**

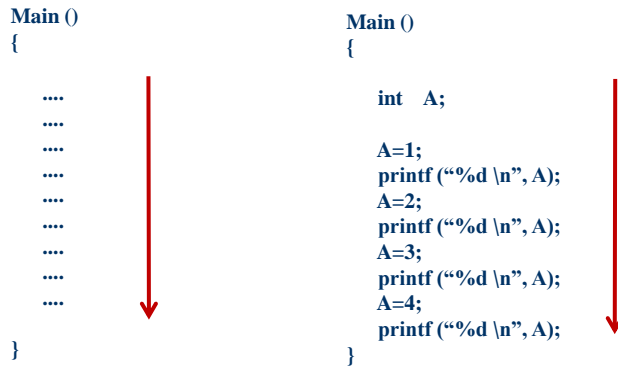
Informações Complementares e Atualizadas:

Consulte REGULARMENTE o material disponível na WIKI

[http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))

Fluxo de um Programa em "C"

Fluxo de Execução de um Programa => **SEM** Repetição (LAÇOS)

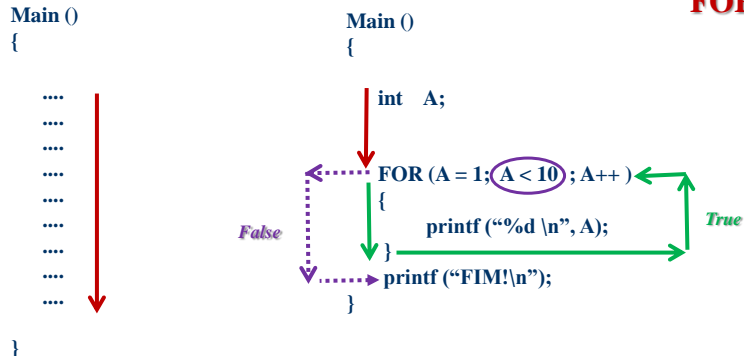


Fluxo Seqüencial: SEM DESVIOS!

Fluxo de um Programa em "C"

Fluxo de Execução de um Programa => **COM** Repetição (LAÇOS)

FOR

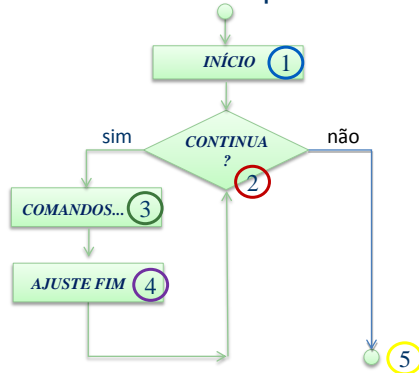


Fluxo Seqüencial: SEM DESVIOS!

Fluxo com Laços: COM DESVIOS!

Comando de Laço: FOR

- Estrutura de laço que permite realizar uma repetição até que uma condição seja satisfeita. Usado para fazer contadores.



```

    1 for (A = 1; 2 A < 10 ; 4 A++)
    {
        printf ("%d \n", A); 3
    }
    printf ("FIM!\n"); 5
    
```

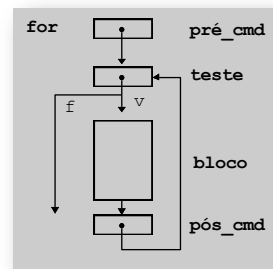
Comando de Repetição: FOR

```

    for (<pré_cmd>; <teste condição> ; <pós_cmd>)
    {
        comandos;
    }
    
```

```

    for (A = 1; A < 10 ; A++)
    {
        printf ("%d \n", A);
    }
    printf ("FIM!\n");
    
```



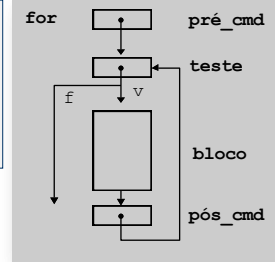
Comando de Laço FOR

Comando de Repetição: FOR

```
for (<pré_cmd>; <teste condição>; <pós_cmd>)  
{  
  comandos;  
}
```

```
A = 1;  
if (A < 10)  
{  
  printf ("%d \n", A);  
  A=A+1;  
  <Laço: Continue no IF>  
}  
printf ("FIM!\n");
```

```
for (A = 1; A < 10 ; A++)  
{  
  printf ("%d \n", A);  
}  
printf ("FIM!\n");
```



Comando de Laço FOR

FOR

```
for ( <expr_inicial>; <condição_de_parada>; <alteração_var_controle> )  
  <comando>;
```

```
for ( contador=0; contador < nro_vezes; contador++ )  
  printf("Contando... %d\n", contador);
```

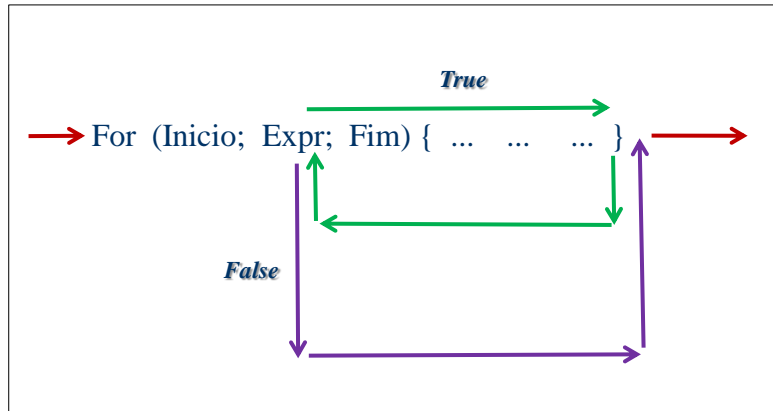
```
for ( contador=10; contador != 0; contador-- )  
{  
  printf("Contagem regressiva...\n");  
  printf("Falta: %d \n", contador);  
}
```

```
CUIDADO: for ( a=1; a <= 10; a++ ) ;  
for ( ; ; ) /* Loop infinito */
```

ERRO!!!

Fluxo de um Programa em "C": FOR

Fluxo de Execução de um Programa => **Repetição FOR**



```
for (<início>; <teste condição final>; <ajuste fim>)  
{  
  comandos;  
}
```

FOR: Para que a variável varie do valor de *inicio* até o valor de *final*

Fluxo de Execução: Laço de Repetição

Comando de Laço: FOR

- Comandos FOR são usados para **contar...**
1, 2, 3, 4, ... N (x = 1 ; x = x + 1; enquanto x <= N)
10, 11, 12, ... M (x = 10; x = x + 1; enquanto x <= M)
10, 9, 8, 7, ... 1 (x = 10; x = x - 1; enquanto x >= 1)
1, 3, 5, 7, 9 ... K (x = 1 ; x = x + 2; enquanto x <= K)
- Comandos FOR são usados **acumular...**
Somar as notas de 10 alunos (somatório)
 $N_1 + N_2 + N_3 + \dots + N_{10}$
- Multiplicar K valores (produtório)
 $V_1 * V_2 * V_3 * \dots * V_k$

$$\sum_{i=1}^K N_i$$

Comando de Laço: FOR – Usos mais comuns

- **CONTAR:** Executar 10 vezes algo (contar até 10)

```
/* Calcula a média simples de 2 notas para uma turma de 10 alunos */  
main()  
{  
    float Nota1, Nota2; Media;  
    int    aluno;  
  
    for (aluno = 1; aluno <= 10; aluno ++)  
    {  
        printf("Digite a Nota1: ");  
        scanf("%f",&Nota1);  
        printf("Digite a Nota2: ");  
        scanf("%f",&Nota2);  
        Media = (Nota1 + Nota2) / 2.0;  
        printf ("Aluno %d - Media = %.2f", aluno, Media);  
    }  
}
```

Comando de Laço: FOR – Usos mais comuns

- **SOMAR: Acumular** o resultado da soma de valores
Por exemplo: somar valores para calculara a média

```
/* Calcula a média geral de toda a turma composta por 30 alunos */  
/* Media: Soma (acumula) as 30 notas e depois divide por 30 */  
main()  
{  
    float Nota, Soma_Total, Media;  
    int    c;  
  
    Soma_Total = 0.0;  
    for (c= 1; c<= 30; c++)  
    {  
        printf("Digite a Nota: ");  
        scanf("%f",&Nota);  
        Soma_Total = Soma_Total + Nota; /* Acumula! */  
    }  
    printf ("Media = %.21f", Soma_Total / 30.0);  
}
```

$$\sum_{i=1}^K N_i$$

Comando de Laço: FOR – Usos mais comuns

- MULTIPLICAR: **Acumular** o resultado do produto de valores
Por exemplo: Fatorial $X*(X-1)*(X-2)*...*1$

```
/* Calcula o fatorial (multiplicações sucessivas) de um número dado */
main()
{
    int Numero, Fatorial, X;

    printf("Digite um nro.: ");
    scanf ("%d",&Numero);

    Fatorial=1;
    for (X=Numero; X >= 1; X--)
    {
        Fatorial=Fatorial * X;
    }

    printf ("Fatorial de %d eh %d\n",Numero,Fatorial);
}
```

$$\prod_{i=1}^K N_i$$

Comando de Laço: FOR

- Comandos FOR podem ser usados aninhados...

Para cada aluno da turma...
 Para cada nota do aluno...

```
for (aluno = 1; aluno <= 30; aluno ++)  
{  
    Total=0;  
    for (aval = 1; aval <= 3; aval ++)  
        {  
            scanf("%lf",&Nota);  
            Total=Total + Nota;  
        }  
    printf ("Aluno %d - Media = %.2lf", aluno, Total/3.0 );  
}
```

Comando de Laço: FOR

- Comandos FOR usados como laço sem fim...

BREAK

break; => Força o término da execução de um loop

```
for ( ; ; )  
{  
    printf ("Digite a Nota do Aluno: ");  
    scanf ("%lf",&Nota);  
    if ((Nota >= 0.0) && (Nota <= 10.0))  
        break;  
    printf("Nota invalida!\n");  
}
```

Comando de Laço: FOR

- Comandos FOR usados como laço sem fim...

CONTINUE

continue; => Força o recomeço da execução de um loop

```
for ( ; ; )  
{  
    printf ("Digite a Nota do Aluno: ");  
    scanf ("%lf",&Nota);  
    if ((Nota < 0.0) || (Nota >10.0))  
        continue;  
    printf ("Nota correta...\n");  
    break;  
}
```


Comando de Controle de Programa **EXIT**

EXIT

`exit;` => Força o término da execução do programa!

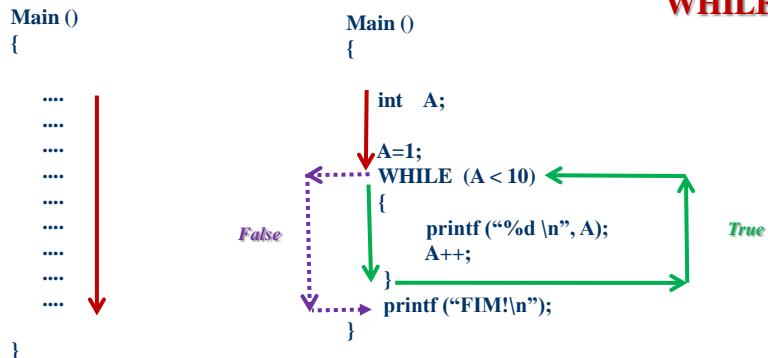
```
printf ("Resposta: ");
scanf ("%d",&valor);
if (valor > LIMITE_MAXIMO)
{
    printf ("Erro: valor inválido!\n");
    printf ("Valor superior ao limite máximo!\n");
    exit(0);
}
```

>> O valor entre parênteses é retornado ao sistema operacional.

Fluxo de um Programa em "C"

Fluxo de Execução de um Programa => **COM Repetição (LAÇOS)**

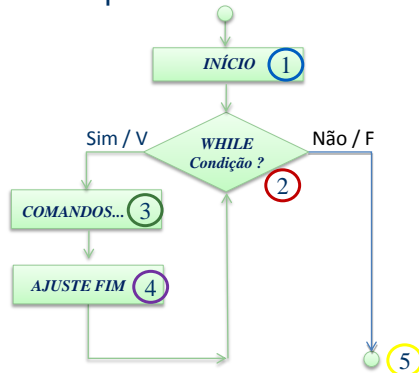
WHILE



Fluxo Seqüencial: SEM DESVIOS!

Comando de Laço: WHILE

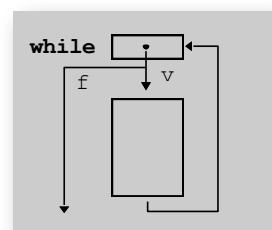
- Estrutura de laço que permite realizar uma repetição até que uma condição se torne falsa. Usado para fazer contadores, acumuladores e repetições



```
A=1; 1
while ( A < 10 ) 2
{
    printf ("%d \n", A); 3
    A++; 4
}
printf ("FIM!\n"); 5
```

Comando de Repetição: WHILE

```
while ( <condição> )
{
    comandos;
}
```



Laço While:

- > 1º avalia condição
- > Se condição é verdadeira, executa comandos do bloco
- > Ao término do bloco, volta a avaliar condição
- > Repete o processo até que condição seja falsa

Comando de Laço WHILE

WHILE

```
while ( <expr_inicial> )  
  <comando>;
```

```
contador = 0;  
while ( contador < nro_vezes )  
  printf("Contando... %d \n", contador++);
```

```
contador = 10;  
printf("Contagem regressiva...\n");  
while ( contador != 0 )  
{  
  printf("Falta: %d \n", contador);  
  contador--;  
}
```

<expr_inicial> é
uma expressão lógica [true, false]

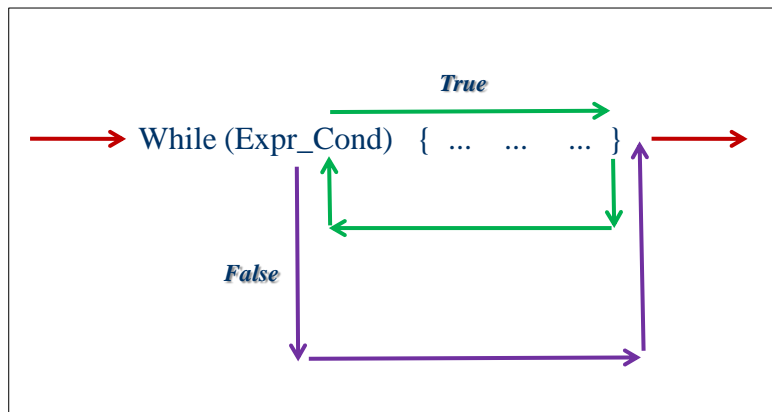
<comando> pode ser único ou
substituído por um bloco { ... }

ERRO!!!

CUIDADO: while (contador < fim) ; contador++;

Fluxo de um Programa em "C"

Fluxo de Execução de um Programa => Repetição WHILE



WHILE (Expr)	[DO]	{ Cmds }
ENQUANTO (Expr)	[FAÇA]	{ Cmds }

Enquanto a expressão for verdadeira
Faça a execução dos comandos

Comando de Laço **WHILE**

WHILE + **BREAK**

BREAK:
Comando para sair da execução
do bloco atual ("quebra fluxo")
{ ... break; ... }

```
double Nota;  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Nota Inválida! Digite novamente... \n");  
}
```

Comando de Laço **WHILE**

WHILE + **BREAK**

```
double Nota;  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota = -1.0;  
  
while (Nota < 0.0 || Nota > 10.0)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota < 0.0 || Nota > 10.0)  
        printf("Erro! Digite novamente... \n");  
}
```

Comando de Laço **DO-WHILE**

DO-WHILE

```
do <comando>  
while ( <expr_inicial> ) ;
```

```
contador = 0;  
do printf ("Contando... %d \n", contador++);  
while ( contador < nro_vezes );
```

```
contador = 10;  
printf("Contagem regressiva...\n");  
do {  
    printf ("Falta: %d \n" , contador);  
    contador--;  
}  
while ( contador != 0 );
```

Comando de Laço **WHILE**

WHILE

DO-WHILE

BREAK

```
double Nota; while  
break  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota = -1.0; while  
  
while (Nota < 0.0 || Nota > 10.0)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota < 0.0 || Nota > 10.0)  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota; do-while  
break  
  
do  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota >= 0.0 && Nota <= 10.0) break;  
    printf("Erro! Digite novamente... \n");  
} while (1)
```

Exercícios – Laço de Repetição

EXERCÍCIOS: USANDO FOR

- Faça um programa que exiba na tela uma contagem regressiva de 10 até 0.
- Faça um programa onde o usuário determine os valores inicial e final de uma contagem (ler os valores inicial e final) e depois realizar uma contagem exibindo os valores na tela.
Nota: O valor inicial pode ser menor que o final (contagem crescente) ou pode ser maior que o final (contagem decrescente).
- Faça um programa que calcule o fatorial de um número fornecido pelo usuário.
Nota: Fatorial de N é $N * (N-1) * (N-2) * \dots * 1 \Rightarrow N!$
Fatorial de 0 é $1 \Rightarrow 0!$ é igual a 1
- Faça um programa que leia 2 notas de cada aluno nas provas P1 e P2. As 2 notas devem ser válidas, ou seja, ter valores entre 0 e 10. Calcule a média ponderada destas 2 notas, onde a nota da P1 tem peso 1 e a nota da P2 tem peso 2, exibindo a média na tela. Repita este procedimento para uma turma de 10 alunos.

Exercícios – Laço de Repetição

EXERCÍCIOS: USANDO WHILE Programas com Repetição (LAÇOS)

Faça um programa que calcule a média de toda uma turma de alunos, baseado na fórmula abaixo...

$$\text{Media} = \frac{\sum_{i=1}^N \text{Nota}_i}{N}$$

N = Número de alunos
 Nota_i = Nota do aluno i
Media = Média dos N alunos

Atenção: Não sabemos quantos alunos tem a turma. O programa deve ir lendo as notas até que o usuário indique que não deseja fornecer mais dados (usando um flag ou uma confirmação explícita).



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO

Web institucional: <http://www.icmc.usp.br/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

Página do Grupo de Pesquisa: <http://www.lrm.icmc.usp.br/>

E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Introdução a Computação – Eng. Ambiental

WIKI - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))

> Programa, Material de Aulas, Critérios de Avaliação,

> Trabalhos Práticos, Datas das Provas, Notas