

Laços de repetição For

Alexsandro Santos Soares
`prof.asoares@gmail.com`

Universidade Federal de Uberlândia
Faculdade de Computação

O laço For

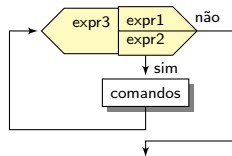
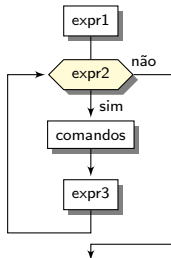
O uso do comando `while` no trecho abaixo

```
expr1;
while (expr2){
    comandos
    expr3;
}
```

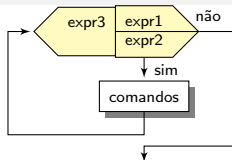
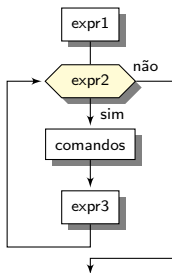
é tão comum em C que foi criado um comando de nome `for` para abreviá-lo:

```
for(expr1; expr2; expr3){
    comandos
}
```

Abaixo à esquerda está o novo símbolo no fluxograma para o **For** e na direita a versão equivalente usando **while**.



O laço For



- A expressão 1 é executada quanto o **for** inicia e, por isso, é chamada de **expressão de inicialização**.
- A expressão 2 é o teste e, como visto no fluxograma equivalente, ela é verificada **antes** de qualquer iteração.
 - Se o teste for falso, os comandos no corpo do laço **for** não serão executados.
- A expressão 3 é a **expressão de atualização** e é executada no final de cada iteração do laço.

O laço For

- Um laço **for** é usado quando um laço precisar ser executado um número conhecido de vezes.
 - Podemos fazer o mesmo com o **while**, mas o laço **for** é mais fácil de ler e mais natural para laços de contagem.

Exemplo 1 (Somando números)

Considere o problema de ler 20 números do teclado e somá-los. Vamos resolver esse problema tanto com **while** quanto com **for**.

```
i = 1;
soma = 0;
while (i <= 20){
    scanf("%d", &a);
    soma += a;
    i++;
} // while
```

```
soma = 0;
for (i = 1; i <= 20; i++){
    scanf("%d", &a);
    soma += a;
} // for
```

Exemplo 2 (Uso de for)

Vamos escrever um programa que pede ao usuário um número inteiro e depois imprime a sequência de números de 1 até o número digitado.

```
11 #include <stdio.h>
12
13 int main(void){
14     int limite = 0;
15
16     printf("Digite o limite da contagem: ");
17     scanf("%d", &limite);
18     for (int i = 1; i <= limite; i++)
19         printf("\t%d\n", i);
20     return 0;
21 } // main
```

O código acima, compilado e executado, produz como saída

```
> ./exemplo2.exe
Digite o limite da contagem: 3
1
2
3
```

Exercício

Exercício 1

Modifique o exemplo anterior e crie programas que

- 1 Imprima somente os números ímpares.
- 2 Imprima os números em ordem decrescente.
- 3 Imprima os números em duas colunas: na primeira ficam os números ímpares e na segunda os pares.

Respostas do Exercício 1

Modifique o exemplo anterior e crie programas que

- 1 Imprima somente os números ímpares.

Mude apenas a atualização do **for**:

```
for (int i = 1; i <= limite; i += 2)
```

Respostas do Exercício 1

Modifique o exemplo anterior e crie programas que

- 1 Imprima somente os números ímpares.

Mude apenas a atualização do **for**:

```
for (int i = 1; i <= limite; i += 2)
```

- 2 Imprima os números em ordem decrescente.

Neste caso, todas as expressões no laço for devem ser alteradas, mas o corpo permanecerá o mesmo:

```
for (int i = limite; i >= 1; i--)
```


Respostas do Exercício 1

Modifique o exemplo anterior e crie programas que

- 1 Imprima somente os números ímpares.

Mude apenas a atualização do `for`:

```
for (int i = 1; i <= limite; i += 2)
```

- 2 Imprima os números em ordem decrescente.

Neste caso, todas as expressões no laço `for` devem ser alteradas, mas o corpo permanecerá o mesmo:

```
for (int i = limite; i >= 1; i--)
```

- 3 Imprima os números em duas colunas: na primeira ficam os números ímpares e na segunda os pares.

Aqui teremos que modificar a expressão de atualização e também o comando `printf`.

```
for (int i = 1; i <= limite; i += 2)  
    printf("\t%2d\t%2d\n", i, i + 1);
```

Laços For aninhados

- Qualquer comando, mesmo um laço **for**, pode ser incluído no corpo de um outro comando **for**.
- Veremos alguns exemplos de uso de laços for aninhados.

Exemplo 3 (For aninhado)

O programa a seguir usa uma laço aninhado para imprimir uma sequência de números em múltiplas linhas.

```
11
12 int main(void){
13
14     for (int i = 1; i <= 3; i++){
15         printf("Linha %d: ", i);
16         for (int j = 1; j <= 5; j++)
17             printf("%3d", j);
18         printf("\n");
19     } // for i
20
21     return 0;
22 } // main
```

A saída produzida é

```
> ./exemplo3.exe
Linha 1:  1 2 3 4 5
Linha 2:  1 2 3 4 5
Linha 3:  1 2 3 4 5
```

A expressão vírgula

- A **expressão vírgula** é uma expressão complexa criada a partir de duas ou mais expressões separadas por vírgulas.
- Embora o seu uso em C seja autorizado em muitos lugares, o uso mais frequente ocorre com laços **for**.
- As expressões são avaliadas da esquerda para a direita. O valor e o tipo final são o valor e o tipo da expressão mais à direita.
 - As demais expressões são usadas apenas devido aos efeitos colaterais.
- O laço *for* do primeiro exemplo pode ser reescrito como

```
for (soma = 0, i = 1; i <= 20; i++){  
    scanf("%d", &a);  
    soma += a;  
} // for
```

- Note que usamos o expressão vírgula para inicializar o acumulador e o contador e depois descartamos o resultado final da expressão.

Exemplo 4 (Juros compostos)

O programa a seguir usa um laço *for* para calcular o valor de retorno de um investimento. Deseja-se saber o valor de um investimento com o passar do tempo, dado um valor inicial e uma taxa anual de juros.

```
11 #include <stdio.h>
12
13 int main(void){
14     double capital=0.0;
15     double montante=0.0;
16     double taxa=0.0;
17     int anos=0;
18     int i = 0;
19
20     printf("Entre com o valor inicial: ");
21     scanf("%lf", &capital);
22     printf("Entre com o taxa de juros anual (nn.n): ");
23     scanf("%lf", &taxa);
24     printf("Entre com o número de anos: ");
25     scanf("%d", &anos);
```

Exemplo 4 – continuação

```
27     printf("\nAno   Valor\n");
28     printf("==== =====\n");
29     for (montante = capital, i = 1;
30         i <= anos;
31         i++){
32         montante *= 1 + taxa/100.0;
33         printf("%3d%11.2lf\n", i, montante);
34     } // for
35
36     return 0;
37 } // main
```

Exemplo 4 – uso

Um exemplo de uso do programa anterior é

```
> ./exemplo4.exe  
Entre com o valor inicial: 10000  
Entre com o taxa de juros anual (nn.n): 7.2  
Entre com o número de anos: 5
```

Ano	Valor
====	=====
1	10720.00
2	11491.84
3	12319.25
4	13206.24
5	14157.09

Exemplo 5 (Triângulo retângulo)

O programa a seguir imprime uma sequência de números na forma de um triângulo retângulo. Pede-se ao usuário para digitar um número com 1 dígito. A partir daí, cada linha, da primeira até a última digitada pelo usuário, imprimirá um sequência de números de um até o número da linha.

Um exemplo de saída é mostrado abaixo para o caso do número digitado ser 6:

```
1
12
123
1234
12345
123456
```


Triângulo retângulo

```
1
12
123
1234
12345
123456
```

Um pseudo código para o programa é mostrado a seguir:

```
1 para linha  $\leftarrow$  1 até limite faça
2   |   para num  $\leftarrow$  1 até linha faça
3     |   escreva num
4   |   fim para
5   |   escreva \n;
6 fim para
```

Exemplo 5 – Triângulo retângulo

```
10 #include <stdio.h>
11
12 int main(void){
13     int limite = 0;
14
15     printf("Entre com um número entre 1 e 9: ");
16     scanf("%d", &limite);
17
18     for (int linha = 1; linha <= limite; linha++){
19         for (int num = 1; num <= linha; num++){
20             printf("%1d", num);
21
22             printf("\n");
23         } // for linha
24
25     return 0;
26 } // main
```

Exemplo 6 (Imprime retângulo)

Agora vamos escrever um programa que imprime o padrão triangular do exemplo anterior, mas que agora completa a linha com asteriscos para formar um retângulo.

```
10 #include <stdio.h>
11
12 int main(void){
13     int limite = 0;
14
15     printf("Entre com um número entre 1 e 9: ");
16     scanf("%d", &limite);
17
18     for (int linha = 1; linha <= limite; linha++){
19         for (int coluna = 1; coluna <= limite; coluna++){
20             if (coluna > linha)
21                 printf("*");
22             else
23                 printf("%1d", coluna);
24
25             printf("\n");
26         } // for linha
27
28         return 0;
29 } // main
```

Exemplo 6 – uso

Um exemplo de uso do programa anterior é

```
> ./exemplo6.exe  
Entre com um número entre 1 e 9: 6  
1*****  
12*****  
123***  
1234**  
12345*  
123456
```

Exemplo 7 (Imprime calendário do mês)

Agora vamos escrever um programa que imprime o calendário de um mês.

A função `imprimeMês` recebe o dia da semana em que cai o primeiro dia do mês – Domingo é 0, Segunda é 1, ..., Sábado é 6 – e o número total de dias do mês.

```

9  #include <stdio.h>
10 void imprimeMes(int primeiroDia, int dias){
11     int diaSemana;
12
13     printf("Dom Seg Ter Qua Qui Sex Sab\n");
14     printf("--- --- --- --- --- --- ---\n");
15
16     for(diaSemana = 0; diaSemana < primeiroDia; diaSemana++){
17         printf("  ");
18
19     for(int diaMes = 1; diaMes <= dias; diaMes++){
20         if (diaSemana > 6) {
21             printf("\n");
22             diaSemana = 1;
23         } else
24             diaSemana++;
25         printf("%3d ", diaMes);
26     } // for
27     printf("\n--- --- --- --- --- --- ---\n");
28     return;
29 } // imprimeMes

```

Exemplo 7 – continuação e uso

```

31 int main(void){
32     imprimeMes(5,30); // 0 primeiro dia do mês caiu na sexta-feira (dia 5)
33     return 0;
34 } // main

```

Um exemplo de uso do programa anterior é

> ./exemplo7.exe

Dom	Seg	Ter	Qua	Qui	Sex	Sab
---	---	---	---	---	---	---
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
---	---	---	---	---	---	---

Exemplo 7 – discussão

- O programa anterior é interessante por duas razões:
 - ① Ele requer dois laços **for**, uma para posicionar a impressão do primeiro dia do mês e outro para imprimir os demais dias.
 - ② A lógica para controlar os dias da semana é simples mas eficiente.
- Examine a inicialização nos dois laços *for*. Como **diaSemana** é usada em ambos os laços, ela não pode ser declarada local ao primeiro laço.
 - Ela deve ser declarada na seção de definições.

Exercício 2

Escreva um programa que crie o seguinte padrão:

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7

1 2 3 4 5 6

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

Exercício 3

Escreva uma **função** que crie o padrão a seguir, dado a altura (número de linhas):

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

Escreva um programa completo para testar a função acima.

Exercício 4

Escreva um programa para imprimir o calendário de um dado ano. O programa deve usar uma nova função de nome `imprimeAno` que recebe o ano e o dia da semana em que cai primeiro dia de janeiro.

Provavelmente será preciso modificar um pouco a função `imprimeMes` e criar uma nova função de nome `ehBissexto` que verificará se o ano é bissexto (note que nesse caso fevereiro terá 29 dias).

Para saber mais

- Forouzan, B. A and Gilbert, R. F. *Computer Science: a structured programming approach using C*. 3rd edition. Cengage Learning, 2007.

- Forouzan, B. A and Gilbert, R. F. *Computer Science: a structured programming approach using C*. 3rd edition. Cengage Learning, 2007.