

Programação de Computadores I

Aula 08

Programação: Estruturas de Repetição

José Romildo Malaquias

Departamento de Computação
Universidade Federal de Ouro Preto

2011-1

Motivação

Como imprimir os três primeiros número ímpares?

```
#include <stdio.h>

int main(void)
{
    printf(" %d\n", 1);
    printf(" %d\n", 3);
    printf(" %d\n", 5);
    return 0;
}
```

Motivação (cont.)

Como imprimir os 100 primeiros número ímpares?

```
#include <stdio.h>

int main(void)
{
    printf(" %d\n", 1);
    printf(" %d\n", 3);
    printf(" %d\n", 5);
    printf(" %d\n", 7);
    printf(" %d\n", 9);
    printf(" %d\n", 11);
    printf(" %d\n", 13);
    // vários comandos depois
    printf(" %d\n", 199);
    return 0;
}
```

Motivação (cont.)

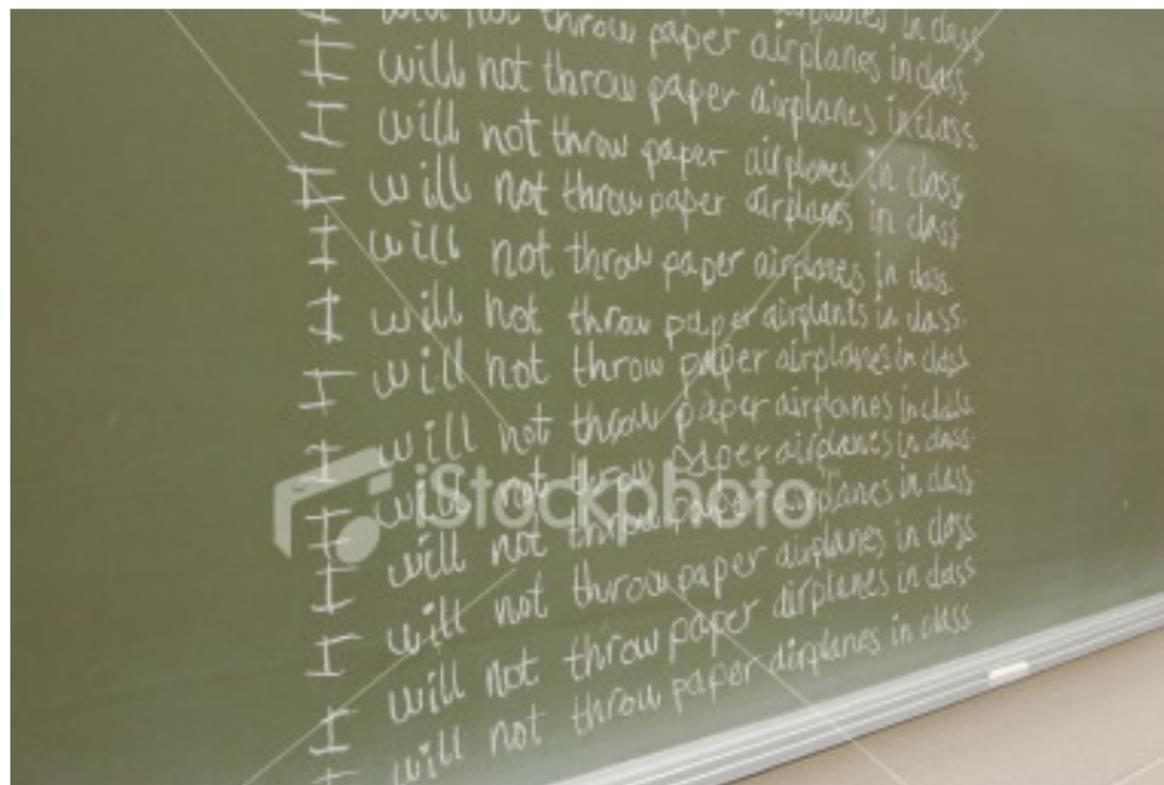
- ▶ As **estruturas de repetição** são um recurso que permite que a execução de um certo segmento do código de um programa seja repetida um certo número de vezes.
- ▶ Na linguagem C existem três estruturas de repetição:
 - o comando `while`,
 - o comando `do while` e,
 - o comando `for`

Motivação (cont.)

Uma professora ordenou a uma aluna que estava jogando aviões de papel durante a aula que ela escrevesse 500 vezes a frase:

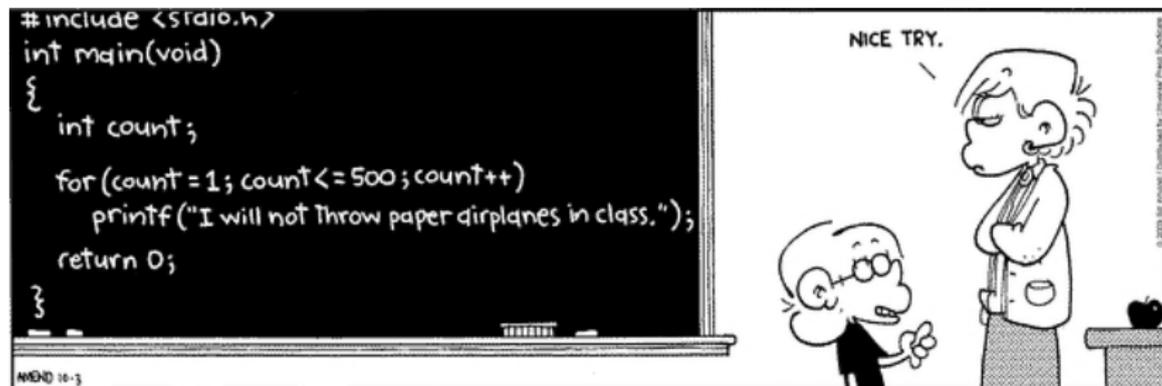
Não devo jogar aviões de papel durante a aula.

Motivação (cont.)



Motivação (cont.)

... mas a menina mais espertinha teve uma idéia melhor ...



Estruturas de repetição

- ▶ São muito comuns as situações em que se deseja repetir um determinado trecho de um programa um certo número de vezes.
- ▶ As estruturas de repetição são muitas vezes chamadas de **laços** ou também de **loops**.
- ▶ O **corpo** da estrutura de repetição é o comando cuja execução deve ser repetida.

Classificação das estruturas de repetição

▶ **Laços contados:**

- Conhecimento prévio de quantas vezes o comando no interior da construção será executado;

▶ **Laços condicionais:**

- Não se conhece de antemão o número de vezes que o conjunto de comandos no interior do laço será repetido.
- Amarrado a uma condição sujeita à modificação pelas instruções do interior do laço.

Estruturas de repetição condicionais

- ▶ O conjunto de comandos em seu interior é executado até que uma determinada condição seja satisfeita.
- ▶ Laços condicionais mais comuns nas linguagens de programação modernas:
 - **enquanto**: laço condicional com teste no início
 - **repita**: laço condicional com teste no final
- ▶ O comando no corpo da estrutura de repetição deve de alguma maneira influenciar na avaliação da condição, de forma que em algum momento ela tenha um valor falso, fazendo com que as repetições sejam encerradas.
- ▶ Geralmente a condição depende de uma variável que está associada a um comando que a atualiza no corpo do laço.
- ▶ Caso isso não ocorra, o programa ficará repetindo indefinidamente este laço, gerando um laço *infinito*, e nunca terminará de um modo normal.

O comando `while`

- ▶ O comando **while** permite que um segmento do programa seja executado repetidas vezes, **enquanto** uma certa condição for **verdadeira**.
- ▶ A forma do comando **while** é a seguinte:

```
while (condição)  
    comando
```

- ▶ Se houver vários comandos a serem repetidos, deve-se inseri-los em um **comando de bloco** no corpo do `while`:

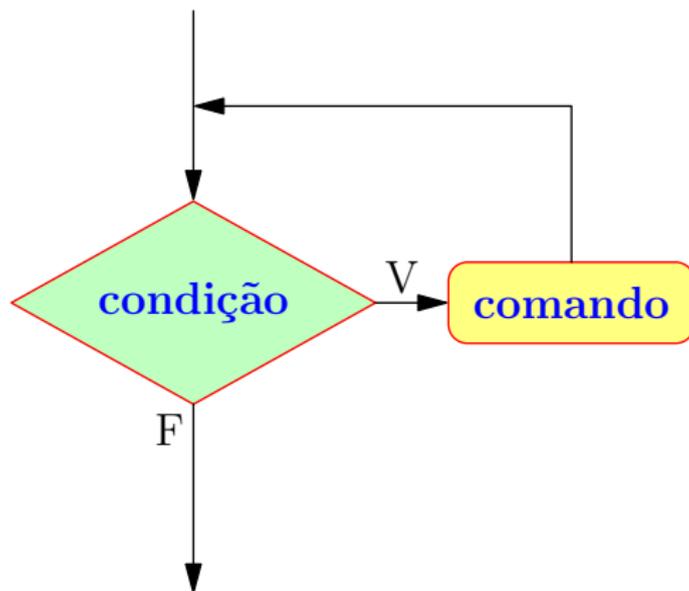
```
while (condição)  
{  
    comando1  
    comando2  
    ...  
    comandon  
}
```

O comando `while` (cont.)

O funcionamento é o seguinte:

1. Avalia a condição.
2. Analisa o seu resultado:
 - 2.1 se o valor da condição for **verdadeiro** então:
 - 2.1.1 executa o comando no corpo do `while`.
 - 2.1.2 volta ao passo 1
 - 2.2 se o valor da condição for **falso** então termina o comando `while`.

O comando while (cont.)



O comando `while` (cont.)

Lembre-se que:

O comando `while` define uma estrutura de repetição na qual:

- ▶ o número de repetições pode não ser conhecido *a priori*,
- ▶ a repetição será controlada pelo valor de uma expressão condicional.

O comando `while`: exemplos

Mostrar os 20 primeiros números naturais positivos:

1, 2, 3, ..., 19, 20

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int num = 1;
    while (num <= 20)
    {
        printf("%d\n", num);
        num = num + 1;
    }
    return 0;
}
```

O comando `while`: exemplos (cont.)

Mostrar os n primeiro números ímpares.

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int impar, i, n;
    printf("Quantidade de impares: ");
    scanf("%d", &n);
    impar = 1;
    i = 1;
    while (i <= n)
    {
        printf("%d\n", impar);
        impar += 2;           // impar = impar + 2
        i++;                 // i = i + 1
    }
    return 0;
}
```

O comando `while`: exemplos (cont.)

Mostrar os n primeiro números ímpares:
(de outra forma)

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int impar, i, n;
    printf("Quantidade de impares: ");
    scanf("%d", &n);
    i = 1;
    while (i <= n)
    {
        impar = 2 * i - 1;
        printf("%d\n", impar);
        i++;
    }
    return 0;
}
```

O comando `while`: exemplos (cont.)

Mostrar os n primeiros números ímpares em ordem decrescente.

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int impar, i, n;
    printf("Quantidade de impares: ");
    scanf("%d", &n);
    i = n;
    while (i >= 1)
    {
        impar = 2 * i - 1;
        printf("%d\n", impar);
        i--;
    }
    return 0;
}
```

O comando `while`: exemplos (cont.)

Escrever um programa que calcula a idade média de um grupo de pessoas. O usuário deverá informar o número de pessoas e em seguida as idades de cada uma delas.

O comando while: exemplos (cont.)

```
#include <stdio.h>
int main(void)
{
    int n, idade;
    printf("Quantidade de pessoas: ");
    scanf("%d", &n);
    double soma = 0.0;
    int contador = 1;
    while (contador <= n)
    {
        printf("Idade da pessoa %d: ", contador);
        scanf("%d", &idade);
        soma = soma + idade;
        contador++;
    }
    double media = soma / n;
    printf("Média: %.2f\n", media);
    return 0;
}
```

O comando `while`: exemplos (cont.)

Escreva um programa em C para calcular o fatorial de um número inteiro e positivo fornecido pelo usuário do programa.

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("Digite um número: ");
    scanf("%d", &num);
    int fat = 1;
    int i = 1;
    while (i <= num)
    {
        fat = fat * i;
        i++;
    }
    printf("O fatorial de %d é %d\n", num, fat);
    return 0;
}
```

O comando `while`: exemplos (cont.)

Escreva uma aplicação que conte a quantidade de números pares e ímpares digitados por um usuário. O usuário pode digitar quantos números quiser, e pode encerrar o programa quando desejar.

O comando while: exemplos (cont.)

```
#include <stdio.h>
int main(void)
{   int num;
    int quantPar = 0;
    int quantImpar = 0;
    char resp = 's';
    while (resp == 's' || resp == 'S')
    {   printf("Digite um número inteiro: ");
        scanf("%d", &num);
        if (num % 2 == 0)
            quantPar++;    // o número é par
        else
            quantImpar++;  // o número é ímpar
        printf("Deseja continuar (s/n)? ");
        scanf(" %c", &resp);
    }
    printf("Quantidade de pares: %d\n", quantPar);
    printf("Quantidade de ímpares: %d\n", quantImpar);
    return 0;
}
```

O comando `while`: exemplos (cont.)

Dado um país A , com 5.000.000 habitantes e uma taxa de crescimento de 3% ao ano, e um país B com 7.000.000 habitantes e uma taxa de crescimento de 2% ao ano. Construa um algoritmo que calcule e imprima quanto tempo é necessário para que a população do país A ultrapasse a população do país B .

O comando while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    double populacaoA = 5000000;
    double populacaoB = 7000000;
    int cont = 0;
    while (populacaoA < populacaoB)
    {
        populacaoA = populacaoA + 0.03 * populacaoA;
        populacaoB = populacaoB + 0.02 * populacaoB;
        cont++;
    }
    printf("São necessários %d anos", cont);
    return 0;
}
```

O comando do `while`

- ▶ Efetua um teste lógico no final de um laço, verificando se é permitido ou não executar novamente o comando no interior do mesmo.
- ▶ Na construção `do-while` o corpo é executado uma ou mais vezes (ou seja, pelo menos uma vez).
- ▶ Na construção `while` o corpo é executado zero ou mais vezes.

O comando do while (cont.)

- ▶ A forma do comando **do while** é:

```
do
    comando
while (condição)
```

- ▶ Se houver vários comandos a serem repetidos, deve-se inseri-los em um **comando de bloco** no corpo do do while:

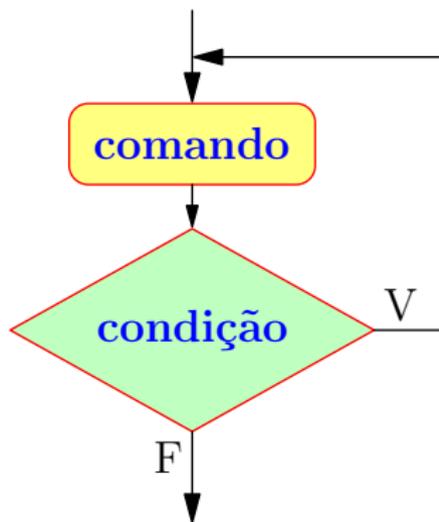
```
do
{
    comando1
    comando2
    ...
    comandon
}
while (condição)
```

O comando do `while` (cont.)

O funcionamento é o seguinte:

1. Executa o comando no corpo do `while`.
2. Avalia a condição.
3. Analisa o seu resultado:
 - 3.1 se o valor da condição for **verdadeiro** então volta ao passo 1
 - 3.2 se o valor da condição for **falso** então termina o comando do `while`.

O comando do while (cont.)



O comando do `while`: exemplos

A conversão de graus fahrenheit para centígrados é obtida pela fórmula

$$C = \frac{5}{9}(F - 32)$$

Escreva um algoritmo que calcule e exiba uma tabela de graus centígrados em função de graus fahrenheit que variem de 150 até 50 de 0,5 em 0,5.

O comando do while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    double F = 150;
    double C;
    printf("%12s %12s\n", "Fahrenheit", "Centigrados")
do
{
    C = 5.0/9.0 * (F - 32.0);
    printf("%12.2f %12.2f\n", F, C);
    F -= 0.5;
} while (F >= 50);
return 0;
}
```

O comando do `while`: exemplos (cont.)

Escrever um programa para ler as idades de um grupo de pessoas e calcular a idade média. A finalização da entrada das idades é dada por um valor negativo (uma idade inválida).

O comando do while: exemplos (cont.)

```
#include <stdio.h>
int main(void)
{
    int idade;
    double soma = 0.0;
    int contador = 0;
    do
    {
        printf("Digite a idade (< 0 para terminar): ");
        scanf("%d", &idade);
        if (idade >= 0)
        {
            soma = soma + idade;
            contador++;
        }
    }
    while (idade >= 0);
    if (contador > 0)
    {
        double media = soma / contador;
        printf("Média: %.2f\n", media);
    }
    return 0;
}
```

O comando do `while`: exemplos (cont.)

Caixa automático:

- ▶ Desenvolver uma aplicação que simula as operações de um caixa automático sobre uma conta bancária.
- ▶ Repetidamente deve ser exibido um menu de opções, e em seguida o usuário deve escolher uma das opções, e a ação correspondente à opção escolhida deve ser executada.
- ▶ As opções do menu são:
 - consultar o saldo
 - depositar um valor
 - sacar um valor
 - sair
- ▶ A execução da aplicação só termina quando o usuário escolher sair.

O comando do while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    double saldo = 0.0;
    double valor;
    int opcao;
    do
    {
        printf("=====\n");
        printf("Banco CEA030\n");
        printf("=====\n");
        printf("\n");
        printf("Menu de opções:\n");
        printf("\n");
        printf("1. Consultar saldo\n");
        printf("2. Depositar\n");
        printf("3. Sacar\n");
        printf("0. Sair\n");
        printf("\n");
        printf("Escolha uma opção: ");
        scanf("%d%c", &opcao);
        printf("\n");
    }
```

O comando do while: exemplos (cont.)

```
switch (opcao)
{
case 0:
    break;
case 1:
    printf("-----\n");
    printf("Consulta do saldo\n");
    printf("-----\n");
    printf("Saldo: %.2f\n", saldo);
    break;
case 2:
    printf("-----\n");
    printf("Depósito\n");
    printf("-----\n");
    printf("Valor: ");
    scanf("%lf%c", &valor);
    if (valor < 0)
        printf("Valor não pode ser negativo! @!##$&!*\n");
    else
        saldo = saldo + valor;
    break;
case 3:
    printf("-----\n");
    printf("Saque\n");
```

O comando do while: exemplos (cont.)

```
printf("-----\n");
printf("Valor: ");
scanf("%lf%c", &valor);
if (valor < 0)
    printf("Valor não pode ser negativo! @!##$&!*\n");
else
    if (valor > saldo)
        printf("Saldo insuficiente! rsssss\n");
    else
        saldo = saldo - valor;
    break;
default:
    printf("Opção inválida!\n");
    printf("Vê se presta mais atenção, seu ... !\n");
}
printf("\n");
if (opcao != 0)
{
    printf("Tecle ENTER para continuar ");
    scanf("%c");
    printf("\n");
}
}
while (opcao != 0);
```

O comando do while: exemplos (cont.)

```
printf("Obrigado pela visita!\n");  
return 0;  
}
```

O comando for

- ▶ **Laços contados** são úteis quando se conhece previamente o número exato de vezes que se deseja executar um determinado comando.
- ▶ O comando **for** é uma estrutura de repetição dotada de mecanismos para contar o número de vezes que o corpo do laço é executado.

O comando for (cont.)

- ▶ A forma do comando for é:

```
for (inicialização; condição; passo)  
    comando
```

- ▶ Se houver vários comandos a serem repetidos, deve-se inseri-los em um **comando de bloco** no corpo do while:

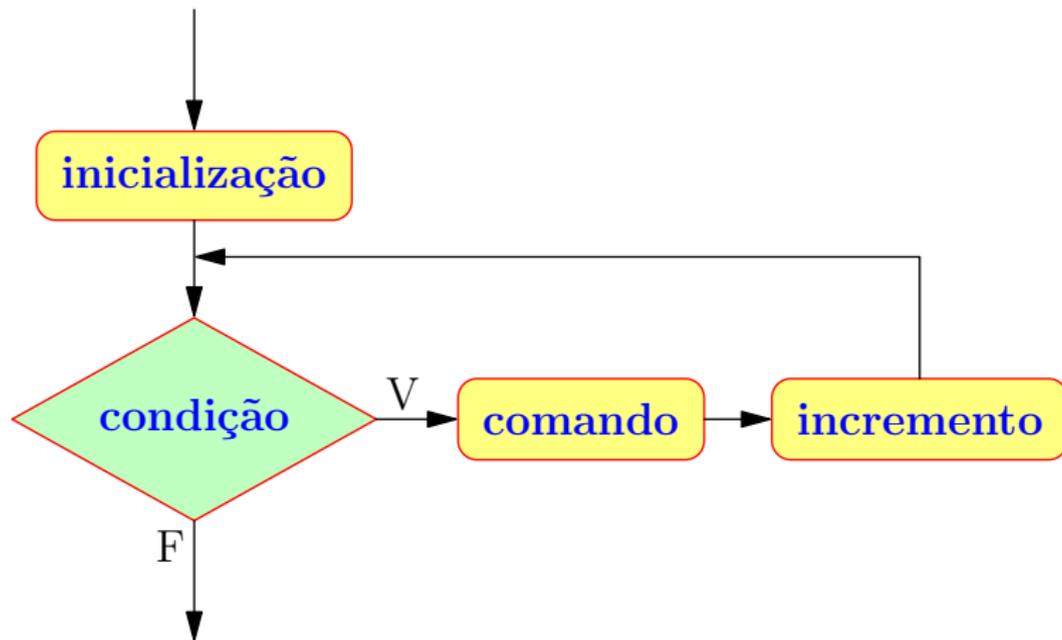
```
for (inicialização; condição; passo)  
{  
    comando1  
    comando2  
    ...  
    comandon  
}
```

O comando for (cont.)

O funcionamento é o seguinte:

1. Executa o comando de inicialização;
2. Testa a condição;
 - 2.1 Se a condição for falsa então o comando `for` termina.
 - 2.2 Se condição for **verdadeira** então
 - 2.2.1 executa o comando no corpo do `for`;
 - 2.2.2 executa o comandos de incremento/decremento;
 - 2.2.3 volta ao passo 2.

O comando for (cont.)



O comando do `while`: exemplos

Mostrar os 10 primeiros números naturais

O comando do while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int i;
    for (i = 0; i < 10; i++)
        printf("%d ", i);
    return 0;
}
```

O comando do `while`: exemplos (cont.)

Mostrar os 10 primeiros número naturais em ordem decrescente

O comando do while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int i;
    for (i = 9; i >=0; i--)
        printf("%d ", i);
    return 0;
}
```

O comando do `while`: exemplos (cont.)

Dado um número inserido por teclado indicar se o número é primo ou não.

O comando do while: exemplos (cont.)

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("Digite um número: ");
    scanf("%d", &num);
    int cont = 0;
    for (int i = 2; i <= num; i++)
        if (num % i == 0)
            cont++;
    if (cont == 1)
        printf("%d é primo\n", num);
    else
        printf("%d não é primo\n", num);
    return 0;
}
```

O comando `break`

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

O comando break: exemplo

```
#include <stdio.h>

int main(void)
{
    for (int n = 10; n > 0; n--)
    {
        printf("%d, ", n);
        if (n == 3)
        {
            printf("contagem regressiva cancelada!\n");
            break;
        }
    }
    return 0;
}
```

Saída na tela:

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

O comando `continue`

- ▶ 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.

O comando continue: exemplo

Por exemplo, saltaremos o número 5 na nossa contagem regressiva:

```
#include <stdio.h>

int main(void)
{
    for (int n = 10; n > 0; n--)
    {
        if (n == 5)
            continue;
        printf("%d, ", n);
    }
    printf("FOGO!\n");
    return 0;
}
```

Saída na tela:

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

FIM

Créditos:

Baseado no material preparado pelo
Prof. Guillermo Cámara-Chávez.