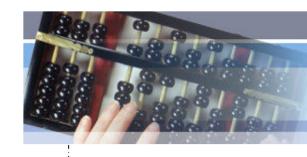
Curso de C

Estruturas de Repetição



Objetivos:

- Aprender:
 - Como vincular a repetição de blocos à condições
 - Quando utilizar cada uma das estruturas disponíveis



Roteiro:

- Introdução
- Comando while
- Comando do...while
- Op. de incremento; formas simplificadas
- Comando for



Introdução:

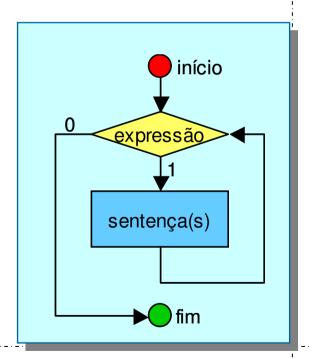
- Estruturas Condicionais:
 - Novidade: Execução condicional de um bloco
- Estruturas de Repetição:
 - Novidade: Repetir a execução de um bloco
 - Controlado por condições
- Objetivo:
 - Preencher uma tabela
 - Aplicar operação a todos elementos da lista
 - Testar vários números
 - Percorrer matrizes, vetores, listas



Estrutura while:

- Executa sentenças enquanto a condição for verdadeira.
- Condição é verificada antes do bloco.

```
início;
while (expressão) {
    sentença;
    sentença;
    ...
}
fim;
```





Exemplo while:

```
Código:
         int numero = 1;
         while (numero <= 10) {</pre>
               printf("%d " , numero);
                numero = numero + 1;
Resultado:
   1 2 3 4 5 6 7 8 9 10
                              EstruturasRepeticao\while01\while01.vcproj
```



Controle das condições:

Inicializa valores usados na condição

Condição que controla repetição

```
int numero = 1;
while (numero <= 10) {
    printf("%d " , numero);
    numero = numero + 1;</pre>
```

Atualiza valores usados na condição



Exemplo while:

```
int main(int argc, char *argv[]) {
       int numero, divisor, resto;
       printf("Digite o numero: ");
       scanf("%d", &numero);
       divisor = 1;
       while (divisor <= numero) {</pre>
              resto = numero % divisor;
               if (resto == 0) {
                      printf("Divisor: %d \n", divisor);
              divisor = divisor + 1;
       return 0;
                            EstruturasRepeticao\Divisores01\Divisores01.vcproj
```



Exemplo while:

```
int main(int argc, char *argv[]) {
       int numeroA, numeroB, resto;
       printf("Digite dois números (ordem crescente): ");
       scanf("%d %d", &numeroA, &numeroB);
       while (numeroA > 0) {
              resto = numeroB % numeroA;
              numeroB = numeroA;
              numeroA = resto;
       printf("MDC: %d", numeroB);
       return 0;
                                   EstruturasRepeticao\mdc01\mdc01.vcproj
```



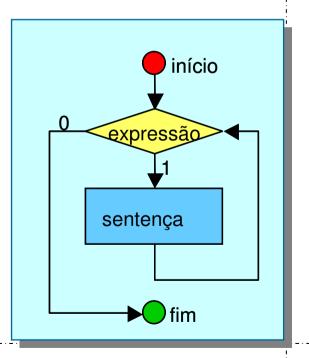
Estrutura while:

Sintaxe simplificada:

- Uma única sentença
- Sem bloco

```
Sintaxe:
```

```
início;
while (expressão)
    sentença;
fim;
```



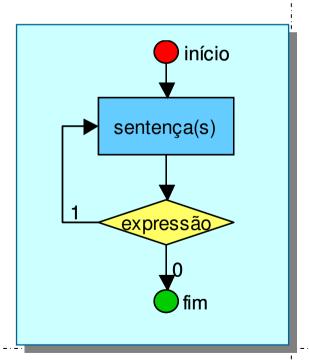


do...while

Estrutura do . . . while:

- Executa sentenças enquanto a condição for verdadeira.
- Condição é verificada depois do bloco

```
início;
do {
    sentença;
    sentença;
    ...
} while (expressão);
fim;
```





do...while

Exemplo do...while:

```
Código:
         int numero = 1;
         do {
                printf("%d " , numero);
                numero = numero + 1;
          } while (numero <= 10);</pre>
Resultado:
   1 2 3 4 5 6 7 8 9 10
                           EstruturasRepeticao\dowhile01\dowhile01.vcproj
```



do...while

Exemplo do...while:

```
int main(int argc, char *argv[]) {
       int numeroA, numeroB, resto;
       printf("Digite dois números (ordem crescente): ");
       scanf("%d %d", &numeroA, &numeroB);
       do {
              resto = numeroB % numeroA;
              numeroB = numeroA;
              numeroA = resto;
       } while (numeroA > 0);
       printf("MDC: %d", numeroB);
       return 0;
                                   EstruturasRepeticao\mdc02\mdc02.vcproj
```



Operadores de incremento:

Antes:

```
numero = numero + 1;
numero = numero - 1;
```

• Agora:

```
++numero;
```

--numero;

 Retornam valor da variável após a operação



Operadores de incremento:

Para:	Atalho:	Original:
Somar um	++numero	numero=numero+1
Subtrair um	numero	numero=numero-1



Operadores de incremento:

Antes:

```
numero = numero + 1;
numero = numero - 1;
```

• Agora:

```
numero++;
numero--;
```

 Retornam valor da variável antes da operação



Operadores de incremento:

Para:	Atalho:	Original:
Somar uma unidade	numero++	numero=numero+1
Subtrair uma unidade	numero	numero=numero-1



Operadores Aritméticos

Operadores aritméticos: notação simplificada

Antes:

```
numero = numero * 10;
numero = numero + 3;
```

Agora:

```
numero *= 10;
numero += 3;
```

Retornam valor da expressão



Operadores de incremento:

Para:	Atalho:	Original:
Somar k unidades	numero += k	numero=numero+k
Subtrair k unidades	numero -= k	numero=numero-k
Multiplicar por k	numero *= k	numero=numero*k
Dividir por k	numero /= k	numero=numero/k



Exemplo:

```
Antes:
               int numero = 1;
               while (numero <= 10) {</pre>
                      printf("%d " , numero);
                      numero = numero + 1;
Depois:
               int numero = 1;
               while (numero <= 10) {</pre>
                      printf("%d " , numero);
                      numero++;
                                    EstruturasRepeticao\while02\while02.vcproj
                                 EstruturasRepeticao\dowhile02\dowhile02.vcproj
```



Controle das condições:

Uma estrutura de repetição tem 4 componentes:

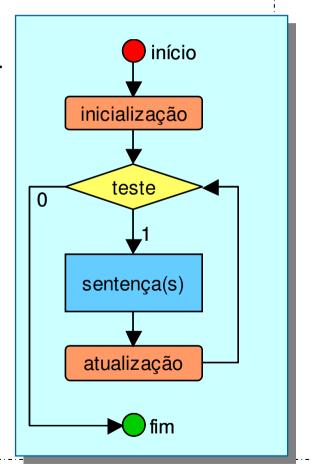


Estrutura for:

• Automatiza estrutura de repetição típica

```
Sintaxe:
```

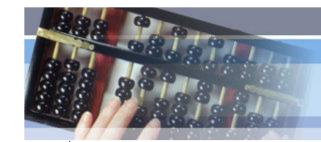
```
início;
for (inicialização;
  teste;
  atualização) {
    sentença;
    sentença;
    ...
}
fim;
```





Exemplo for:

```
Código:
int numero;
for (numero = 1; numero <= 10; numero++) {</pre>
      printf("%d ", numero);
Resultado:
   1 2 3 4 5 6 7 8 9 10
                                EstruturasRepeticao\for01\for01.vcproj
```



Por que usar for?

- Cabeçalho agrupa:
 - Inicialização
 - Condição
 - Atualização

Programador não "esquece" nenhuma etapa

- Separa:
 - Controle (lógica) de repetição
 - Código a ser repetido

Código organizado



Exemplo for:

```
int main(int argc, char *argv[]) {
    int numero, divisor, resto;
    printf("Digite o numero: ");
    scanf("%d", &numero);
    for (divisor = 1; divisor <= numero; divisor++) {</pre>
         resto = numero % divisor;
         if (resto == 0) {
              printf("Divisor: %d \n", divisor);
    return 0;
                             EstruturasRepeticao\Divisores02\Divisores02.vcproj
```



Casos de Uso

Casos de Uso:

- while (expressão) { ... }
 - Não há variável contadora
 - Inicialização, teste ou atualização complexos
 - Informações da condição obtidas na execução
- do { ... } while (expressão);
 - Executar um bloco pelo menos uma vez
 - Só é possível avaliar a condição depois de executar
 - Informações da condição obtidas após execução



Casos de Uso

Casos de Uso:

- for (inicialização; teste; reinicialização) { ... }
 - Há variável contadora de repetições
 - Inicialização, teste e atualização simples
 - Separa claramente as instruções de controle das instruções do bloco

Exemplos



Caso 1: for:

```
int main(int argc, char *argv[]) {
   int quantidade, contador;
  double valor, soma = 0, media;
  printf("Quantidade de valores: ");
   scanf("%d", &quantidade);
   for (contador = 1; contador <= quantidade; contador++) {</pre>
     printf("Valor: ");
     scanf("%lf", &valor);
     soma += valor;
  printf("Media: %f", soma / quantidade);
   return 0;
                                    EstruturasRepeticao\Caso01\Caso01.vcproj
```



Caso 2: while:

```
int main(int argc, char *argv[]) {
  int quantidade, contador;
  double valor, soma = 0, media;
  printf("Quantidade de valores: ");
  scanf("%d", &quantidade);
  contador = 1;
  while (contador <= quantidade) {</pre>
     printf("Valor: ");
     scanf("%lf", &valor);
     soma += valor;
     contador++;
  printf("Media: %f", soma / quantidade);
  return 0;
                           EstruturasRepeticao\Caso02\Caso02.vcproj
```



Caso 3: while:

```
int main(int argc, char *argv[]) {
  int quantidade = 0;
  double valor, soma = 0, media;
  printf("Escreva valores. -1 termina.\n");
  scanf("%lf", &valor);
  while (valor >= 0.0) {
     soma += valor;
     quantidade++;
     scanf("%lf", &valor);
  printf("Media: %f", soma / quantidade);
  return 0;
                           EstruturasRepeticao\Caso03\Caso03.vcproj
```



Caso 4: do...while:

```
int main(int argc, char *argv[]) {
  int quantidade = 0;
  double valor, soma = 0, media;
  printf("Escreva valores. -1 termina.\n");
  do {
     scanf("%lf", &valor);
     if (valor >= 0.0) {
       soma += valor;
       quantidade++;
   } while (valor >= 0.0);
  printf("Media: %f", soma / quantidade);
  return 0;
                           EstruturasRepeticao\Caso04\Caso04.vcproj
```