Comandos de Repetição

DCC 119 – Algoritmos

Uso de acumulador

- Acumulador é uma variável utilizada para armazenar a soma (ou o produto) de uma sequência de valores.

- A variável usada como acumulador recebe um acréscimo a cada iteração, isto é, seu valor anterior é usado em sua atualização ao longo das iterações.

```
soma = soma + novo_valor;
```

- A variável usada como **acumulador** precisa ser inicializada antes do laço.

```
soma = 0;
produto = 1;
```

Uso de acumulador

Desenvolva um algoritmo que leia uma sequência de números inteiros, calcule e imprima a soma dos valores lidos. A sequência deve terminar quando o número 0 (zero) for lido.

Uso de acumulador

Desenvolva um algoritmo que leia uma sequência de números inteiros, calcule e imprima a soma dos valores lidos. A sequência deve terminar quando o número 0 (zero) for lido.

soma → o valor da variável começa com zero (elemento neutro da adição) ;

→ a cada iteração, conserva o valor anterior com acréscimo de um novo valor;

→ ao final de cada iteração, o valor da variável contém a soma parcial dos elementos ;

→ ao final do laço, o valor da variável contém a soma dos elementos.

Uso de acumulador

```c
#include <stdio.h>
int main()
{
   int num, soma;
   soma = 0; // inicializa acumulador
   printf("Digite um numero inteiro: ");
   scanf("%d", &num);
   while( num != 0 )
   {
      soma = soma + num; //atualiza acumulador
      printf("Soma parcial: %d",soma);
      printf("\nDigite um numero inteiro: ");
      scanf("%d", &num);
   }
   printf("Soma total: %d",soma);
   return 0;
}
```

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro:
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
Soma parcial: 7
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
Soma parcial: 7
Digite um numero inteiro:
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
Soma parcial: 7
Digite um numero inteiro: 0
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	
13	0	7	

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
Soma parcial: 7
Digite um numero inteiro: 0
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	
13	0	7	
8	0	7	F

Uso de acumulador

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

```
Digite um numero inteiro: 9
Soma parcial: 9
Digite um numero inteiro: -2
Soma parcial: 7
Digite um numero inteiro: 0
Soma total: 7
```

TESTE DE MESA

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
13	-2	9	
8	-2	9	V
10	-2	7	
13	0	7	
8	0	7	F
15	0	7	

Uso de acumulador

Acumuladores precisam de:

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

Inicialização

Variável precisa ser inicializada (normalmente com elemento neutro).

Atualização

Variável precisa conservar o valor acumulado e acrescentar novo valor.

Uso de acumulador

Impressão de acumuladores:

```
1  #include <stdio.h>
2  int main()
3  {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10       soma = soma + num; //atualiza acumulador
11       printf("Soma parcial: %d",soma);
12       printf("\nDigite um numero inteiro: ");
13       scanf("%d", &num);
14    }
15    printf("Soma total: %d",soma);
16    return 0;
17 }
```

Valor parcial

Impressão da variável é feita no laço, após sua atualização.

Valor total

Impressão da variável é feita após o laço.

Exercício

1. Escreva uma função que retorne o valor total a ser pago em uma compra de supermercado. A função deve ler a quantidade e o preço unitário de cada produto, atualizando o subtotal a cada iteração do laço. Considere que a compra será encerrada quando for digitado um valor menor ou igual a zero para a quantidade de um produto.

   Faça um programa para chamar e imprimir o resultado

   sua função.

   Teste seu programa com a sequência:

   ```
   1  4.02  3  7.56  1  5.3  -4
   ```

   ou, dependendo de sua implementação,

   ```
   1  4.02  3  7.56  1  5.3  -4  2.0
   ```

Cálculo de média

Desenvolva um algoritmo que leia uma sequência de números inteiros e imprima a media aritmética dos valores lidos. A sequência deve terminar quando o número 0 (zero) for lido.

Cálculo de média

Desenvolva um algoritmo que leia uma sequência de números inteiros e imprima a media aritmética dos valores lidos. A sequência deve terminar quando o número 0 (zero) for lido.

media → a média aritmética corresponde à soma dos valores dividida pelo número de valores;

→ um acumulador e um contador serão necessários neste cálculo;

→ o resultado da divisão deve ser um valor real, mesmo se os valores envolvidos são inteiros;

→ a divisão deve ser realizada após o fim do laço, quando os valores do contador e do acumulador já não sofrerão alterações.

Cálculo de média

Desenvolva um algoritmo que leia uma sequência de números inteiros e imprima a media aritmética dos valores lidos. A sequência deve terminar quando o número 0 (zero) for lido.

media → a média aritmética corresponde à soma dos valores dividida pelo número de valores;

→ um acumulador e um contador serão necessários neste cálculo;

→ o resultado da divisão deve ser um valor real, mesmo se os valores envolvidos são inteiros;

→ a divisão deve ser realizada após o fim do laço, quando os valores do contador e do acumulador já não sofrerão alterações.

Cálculo de média

```
1   #include <stdio.h>
2   int main()
3   {
4     int num, soma, cont;
5     float media;
6     soma = 0;              // inicializa acumulador
7     cont = 0;              // inicializa contador
8     printf("Digite um numero inteiro: ");
9     scanf("%d", &num);
10    while( num != 0 )
11    {
12      soma = soma + num; //atualiza acumulador
13      cont++;             //atualiza contador
14      printf("\nDigite um numero inteiro: ");
15      scanf("%d", &num);
16    }
17    media = soma / (float) cont; //conversao para divisao real
18    printf("\nMedia dos elementos: %f",media);
19    return 0;
20  }
```

Outro exemplo: soma de dígitos

Desenvolva um algoritmo que leia um número inteiro positivo e imprima a soma de seus dígitos.

Outro exemplo: soma de dígitos

Desenvolva um algoritmo que leia um número inteiro positivo e imprima a soma de seus dígitos.

→ a separação dos dígitos de um número pode ser feita através das seguintes operações:

- o resto da divisão por 10 permite que o dígito menos significativo seja obtido:

```
530479 % 10 → 9
```

- a divisão inteira por 10 permite que o dígito menos significativo seja descartado e o dígito seguinte assuma esta posição:

```
530479 / 10 → 53047
```

Outro exemplo: soma de dígitos

Desenvolva um algoritmo que leia um número inteiro positivo e imprima a soma de seus dígitos.

→ a separação dos dígitos de um número pode ser feita através das seguintes operações:

```
530479 % 10 → 9
530479 / 10 → 53047
53047 % 10 → 7
53047 / 10 → 5304
5304 % 10 → 4
5304 / 10 → 530
```

```
530 % 10 → 0
530 / 10 → 53
53 % 10 → 3
53 / 10 → 5
5 % 10 → 5
5 / 10 → 0
```

Outro exemplo: soma de dígitos

Desenvolva um algoritmo que leia um número inteiro positivo e imprima a soma de seus dígitos.

→ a separação dos dígitos de um número pode ser feita através das seguintes operações:

`530479 % 10 → 9`	`digito = num % 10;`
`530479 / 10 → 53047`	`num = num / 10;`
`53047 % 10 → 7`	`digito = num % 10;`
`53047 / 10 → 5304`	`num = num / 10;`
`5304 % 10 → 4`	`digito = num % 10;`
`5304 / 10 → 530`	`num = num / 10;`
`...`	`...`

Outro exemplo: soma de dígitos

```c
#include <stdio.h>
 int main()
 {
   int num, digito, soma;
   soma = 0; // inicializa acumulador
   printf("Digite um numero inteiro: ");
   scanf("%d", &num);
   while( num != 0 )
   {
     digito = num % 10;     // obtem digito menos significativo
     soma = soma + digito; // acrescenta o valor do digito a soma
     num = num / 10;        // descarta o digito armazenado
   }
   printf("\nSoma dos digitos: %d",soma);
   return 0;
}
```

Outro exemplo: soma de dígitos

```
1  #include <stdio.h>
2   int main()
3   {
4     int num, digito, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um numero inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10      digito = num % 10;     // obtem digito menos significativo
11      soma = soma + digito; // acrescenta o valor do digito a soma
12      num = num / 10;        // descarta o digito armazenado
13    }
14    printf("\nSoma dos digitos: %d",soma);
15    return 0;
16  }
```

Atualização da variável da condição

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

O uso do comando **`for`** é muito similar ao do comando **`while`**.

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

Funcionalmente, **`for`** e **`while`** são idênticos, apenas a sintaxe dos comandos muda.

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

O **for** concentra os comandos de inicialização, condição e atualização entre parênteses, separados por ponto e vírgula.

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

inicialização

O comando de inicialização do **`for`** é realizado uma única vez, antes de sua primeira execução.

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

condição

A condição do **`for`** é testada no início de toda iteração. O bloco de comandos interno só é executado se a condição for VERDADEIRA.

Repetição com variável de controle

```
for ( inicializacao; condicao; atualizacao )
{
  blocoDeComandos1;
}
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

atualização

A atualização é realizada após a execução do bloco de comandos interno e antes do teste da condição da iteração seguinte.

Repetição com variável de controle

```
for( contador=0; contador < 5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

Em geral, **`for`** é mais usado quando há uma variável de controle, como um contador na condição.

Repetição com variável de controle

```
for( contador=0; contador < 5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

Em geral, **`for`** é mais usado quando há uma variável de controle, como um contador na condição.

Repetição com variável de controle

```
for( contador=0; contador < 5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

Em geral, **`for`** é mais usado quando há uma variável de controle, como um contador na condição.

Repetição com variável de controle

```
for( contador=0; contador < 5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

Em geral, **`for`** é mais usado quando há uma variável de controle, como um contador na condição.

Repetição com variável de controle

```
for( contador=0; contador<5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

A vantagem do **`for`** neste caso é concentrar inicialização, condição e atualização, evitando que algum destes comandos seja esquecido.

Repetição com variável de controle

```
for( contador=0; contador<5; contador++ )
{
   blocoDeComandos1;
}
blocoDeComandos2;
```

```
contador = 0;
while ( contador < 5 )
{
    blocoDeComandos1;
    contador++;
}
blocoDeComandos2;
```

O comando **`while`** é mais indicado quando o controle do laço envolve FLAG ou condições complexas, uma vez que o código fica mais claro de ser entendido.

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

TESTE DE MESA

linha	i	teste	
4	?		int i;

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1  2
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1  2
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`
5	3	---	`i++;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1  2
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`
5	3	---	`i++;`
5	3	V	`i<=3;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
1  2  3
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`
5	3	---	`i++;`
5	3	V	`i<=3;`
7	3	---	`printf(" %d ",i);`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
 1   2   3
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`
5	3	---	`i++;`
5	3	V	`i<=3;`
7	3	---	`printf(" %d ",i);`
5	4	---	`i++;`

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4    int i;
 5    for( i=1; i<=3; i++ )
 6    {
 7       printf(" %d ",i);
 8    }
 9    return 0;
10 }
```

```
 1   2   3
```

TESTE DE MESA

linha	i	teste	
4	?	---	int i;
5	1	---	i=1;
5	1	V	i<=3;
7	1	---	printf(" %d ",i);
5	2	---	i++;
5	2	V	i<=3;
7	2	---	printf(" %d ",i);
5	3	---	i++;
5	3	V	i<=3;
7	3	---	printf(" %d ",i);
5	4	---	i++;
5	4	F	i<=3;

Repetição com variável de controle

Detalhe da ordem de execução das instruções

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4   int i;
 5   for( i=1; i<=3; i++ )
 6   {
 7     printf(" %d ",i);
 8   }
 9   return 0;
10 }
```

```
1  2  3
```

TESTE DE MESA

linha	i	teste	
4	?	---	`int i;`
5	1	---	`i=1;`
5	1	V	`i<=3;`
7	1	---	`printf(" %d ",i);`
5	2	---	`i++;`
5	2	V	`i<=3;`
7	2	---	`printf(" %d ",i);`
5	3	---	`i++;`
5	3	V	`i<=3;`
7	3	---	`printf(" %d ",i);`
5	4	---	`i++;`
5	4	F	`i<=3;`
9	4	---	`return 0;`

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

Desenvolva um algoritmo que imprima todos os valores inteiros em um intervalo indicado pelo usuário.

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

Desenvolva um algoritmo que imprima todos os valores inteiros em um intervalo indicado pelo usuário.

→ o usuário vai indicar o valor inicial e o valor final do intervalo;

→ todos os valores do intervalo devem ser impressos, um a um;

→ um contador pode ser utilizado para indicar cada valor a ser impresso.

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1  #include <stdio.h>
2  int main()
3  {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12       printf(" %d",cont);
13    }
14    return 0;
15 }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo:
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo:
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1  #include <stdio.h>
2  int main()
3  {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12       printf(" %d",cont);
13    }
14    return 0;
15 }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	**4**	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1  #include <stdio.h>
2  int main()
3  {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12       printf(" %d",cont);
13    }
14    return 0;
15 }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	
10	3	5	5	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	
10	3	5	5	V

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1  #include <stdio.h>
2  int main()
3  {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12       printf(" %d",cont);
13    }
14    return 0;
15 }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	
10	3	5	5	V
10	3	5	6	

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	
10	3	5	5	V
10	3	5	6	
10	3	5	6	F

Repetição com variável de controle

Exemplo 1: Imprime valores em um intervalo

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     printf("Intervalo: ");
10    for( cont=inicio; cont<=fim; cont++ )
11    {
12      printf(" %d",cont);
13    }
14    return 0;
15  }
```

```
Digite o menor numero: 3
Digite o maior numero: 5
Intervalo: 3 4 5
```

TESTE DE MESA

linha	inicio	fim	cont	teste
4	?	?	?	
6	3	?	?	
8	3	5	?	
10	3	5	3	
10	3	5	3	V
10	3	5	4	
10	3	5	4	V
10	3	5	5	
10	3	5	5	V
10	3	5	6	
10	3	5	6	F

Repetição com variável de controle

Exemplo 2: Imprime tabuada

Desenvolva um algoritmo que leia dois números inteiros **`tab`** e **`limite`** e imprima a tabuada de **`tab`** desde 1 até **`limite`**.

Repetição com variável de controle

Exemplo 2: Imprime tabuada

Desenvolva um algoritmo que leia dois números inteiros **`tab`** e **`limite`** e imprima a tabuada de **`tab`** desde 1 até **`limite`**.

→ o usuário vai indicar os valores tab e limite;

→ todos os múltiplos de tab devem ser impressos, um a um, com multiplicadores variando de 1 a limite;

→ um contador pode ser utilizado para armazenar os multiplicadores.

Repetição com variável de controle

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13              tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13              tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13              tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	
10	5	3	3	V

Exemplo 2: Imprime tabuada

```
1  #include <stdio.h>
2  int main()
3  {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12       printf("\n%2d x %2d = %2d",
13              tab,cont,cont*tab);
14    }
15    return 0;
16 }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
 5 x  3 = 15
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	
10	5	3	3	V

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
 5 x  3 = 15
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	
10	5	3	3	V
10	5	3	4	

Exemplo 2: Imprime tabuada

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,tab,limite;
5     printf("Tabuada de: ");
6     scanf("%d", &tab);
7     printf("e o ultimo multiplicador: ");
8     scanf("%d", &limite);
9     printf("Tabuada de %d: ",tab);
10    for( cont=1; cont<=limite; cont++ )
11    {
12      printf("\n%2d x %2d = %2d",
13             tab,cont,cont*tab);
14    }
15    return 0;
16  }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
 5 x  3 = 15
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	
10	5	3	3	V
10	5	3	4	
10	5	3	4	F

Exemplo 2: Imprime tabuada

```
1  #include <stdio.h>
2  int main()
3  {
4    int cont,tab,limite;
5    printf("Tabuada de: ");
6    scanf("%d", &tab);
7    printf("e o ultimo multiplicador: ");
8    scanf("%d", &limite);
9    printf("Tabuada de %d: ",tab);
10   for( cont=1; cont<=limite; cont++ )
11   {
12     printf("\n%2d x %2d = %2d",
13            tab,cont,cont*tab);
14   }
15   return 0;
16 }
```

```
Tabuada de: 5
e o ultimo multiplicador: 3
Tabuada de 5:
 5 x  1 =  5
 5 x  2 = 10
 5 x  3 = 15
```

TESTE DE MESA

linha	tab	limite	cont	teste
4	?	?	?	
6	5	?	?	
8	5	3	?	
10	5	3	1	
10	5	3	1	V
10	5	3	2	
10	5	3	2	V
10	5	3	3	
10	5	3	3	V
10	5	3	4	
10	5	3	4	F

Exercício

2. Elabore uma função que receba o valor n por parâmetro. A função deve ler n valores reais e retornar sua média. Use obrigatoriamente o comando for.

   Faça um programa que leia um valor inteiro n, chame a função acima e imprima o resultado obtido.

   Teste seu programa com os valores:

   ```
   5  5.0  7.5  8.5  7.0  5.5
   ```

3. Faça uma função que receba um valor n e retorne o seu fatorial. Faça um programa para testar sua função.

   Teste seu programa com o valor:

   ```
   4
   ```

Repetição com teste no final

```
do
{
   atualizacao;
   blocoDeComandos1;
} while ( condicao );
blocoDeComandos2;
```

O comando **do-while** é similar aos comandos **while** e **for**, mas, neste caso, o teste da condição acontece em momentos distintos.

Repetição com teste no final

```
do
{
  atualizacao;
  blocoDeComandos1;
} while ( condicao );
blocoDeComandos2;
```

```
inicializacao;
while ( condicao )
{
  blocoDeComandos1;
  atualizacao;
}
blocoDeComandos2;
```

Como **do-while** só testa a condição após a primeira execução do bloco de comandos, este bloco certamente será executado ao menos uma vez no programa.

Repetição com teste no final

Exemplo 1: Imprime o quadrado de 15 inteiros

Desenvolva um algoritmo que leia 15 números inteiros e imprima o quadrado de cada um deles.

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont, num;
5     cont = 0; //inicializa contador
6     do
7     {
8       printf("\nDigite o %do numero: ",cont);
9       scanf("%d", &num);
10      printf("\nQuadrado de %d: %d ",
11      num, num*num );
12      cont++;
13    } while ( cont < 15 );
14    return 0;
15  }
```

Repetição com teste no final

Faça o teste de mesa dos dois programas.

Qual a diferença entre eles?

```c
#include <stdio.h>
int main()
{
   int cont;
   cont = 0;
   do
   {
      printf("%d",cont);
      cont++;
   } while ( cont < 2 );
   return 0;
}
```

```c
#include <stdio.h>
int main()
{
   int cont;
   cont = 0;
   while ( cont < 2 )
   {
      printf("%d",cont);
      cont++;
   }
   return 0;
}
```

Repetição com teste no final

Exemplo 2: Imprime valores em um intervalo

Desenvolva um algoritmo que imprima todos os valores inteiros em um intervalo indicado pelo usuário.

```
1   #include <stdio.h>
2   int main()
3   {
4     int cont,inicio,fim;
5     printf("Digite o menor numero: ");
6     scanf("%d", &inicio);
7     printf("Digite o maior numero: ");
8     scanf("%d", &fim);
9     cont = inicio;
10    printf("\nIntervalo: ");
11    do
12    {
13      printf(" %d ",cont);
14      cont++;
15    } while( cont <= fim );
16    return 0;
17  }
```

O comando `break`

Já vimos o comando **`break`** com o **`switch`**:

O **`break`** faz com que o fluxo de execução saia do **`switch`** e continue a execução logo após o bloco do **`switch`**.

O comando `break`

Da mesma forma, o comando **`break`** pode ser utilizado dentro de qualquer laço:

O **`break`** faz com que o fluxo de execução saia do laço e a execução continue logo após o fim do laço.

Assim, o **`break`** pode ser usado dentro do bloco de comandos do **`while`**, do **`for`** ou do **`do-while`**.

O comando `break`

Exemplo: Retirada de valores limitada

Um banco permite que seus clientes com conta bancária simples façam até 5 retiradas por mês. Mas estes clientes não tem crédito pré-aprovado e, a retirada deve ser limitada ao valor contido no saldo inicial.

O comando `break`

Exemplo: Retirada de valores limitada

Um banco permite que seus clientes com conta bancária simples façam até 5 retiradas por mês. Mas estes clientes não tem crédito pré-aprovado e, a retirada deve ser limitada ao valor contido no saldo inicial.

→ o usuário vai indicar o saldo inicial;

→ o usuário vai digitar o valor de cada retirada, se o total retirado exceder o saldo inicial, a sequência deve ser interrompida.

O comando `break`

Exemplo: Retirada de valores limitada

Um banco permite que seus clientes com conta bancária simples façam até 5 retiradas por mês. Mas estes clientes não tem crédito pré-aprovado e, a retirada deve ser limitada ao valor contido no saldo inicial.

O comando `break`

Exemplo: Retirada de valores limitada

```
 1 #include <stdio.h>
 2 int main()
 3 {
 4   int cont = 0;
 5   float saldo, retirada;
 6   printf("Digite o saldo inicial:  R$");
 7   scanf("%f", &saldo);
 8   for( cont = 0; cont < 5; cont++ ) {
 9      printf("Digite a %da retirada:  R$",cont + 1);
10      scanf("%f", &retirada);
11      if( saldo - retirada < 0 ) {
12         printf("Só é possível retirar %f.", saldo);
13         saldo = 0;
14         break;
15      }
16      saldo = saldo - retirada;
17   }
18   printf("\nSaldo final: %f", saldo);
19   return 0;
20 }
```

Exercícios

4. Faça um programa que imprima todos os números **pares** no intervalo decrescente de 100 a 1.

5. Faça um programa que leia um inteiro positivo x e imprima todas as potências de 2 no intervalo entre 0 e x. Use uma variável acumuladora para calcular a potência de 2.
   Teste seu programa com o valor: `21`

6. Faça um programa que leia um número inteiro e positivo e verifique se este é ou não um número primo.
   Teste seu programa com o valor: `7`

7. Modifique o programa anterior para que a leitura seja repetida **enquanto o valor digitado for inválido**, isto é, até que o valor seja inteiro e positivo.
   Teste seu programa com os valores:
   `-5  -10  4`

Exercícios

8. Dada uma dívida de R$10.000,00 que cresce a juros de 2,5% ao mês e uma aplicação de R$ 1.500,00 com rendimento de 4% ao mês, escreva um algoritmo que determine o número de meses necessários para que a aplicação seja suficiente para pagar a dívida.

9. Elabore um algoritmo que calcule o valor de S:

$$S = \frac{2}{50} + \frac{2^2}{48} + \frac{2^3}{46} + \cdots + \frac{2^{25}}{2}$$

Exercícios

DESAFIO: Uma expressão em C pode conter parênteses para identificar a ordem de execução desejada pelo programador. Um dos erros que o compilador pode identificar quando compilamos um programa está relacionado a expressões com parênteses não balanceados. Note que, para cada parêntese aberto em uma expressão, obrigatoriamente deve existir um parêntese que o fecha, de forma a deixar o código compilável. Faça um programa em C que leia caracteres informados pelo usuário enquanto o caractere digitado for '(' ou ')'. Assim que um caractere diferente de '(' e ')' for digitado, você deve imprimir "Balanceados!", se os parênteses estiverem balanceados, ou "Não balanceados!", caso contrário, e o programa deve ser finalizado. Exemplos:

(( ) → Não balanceados!

) (→ Não balanceados!

((( )) ( )) → Balanceados!

(( ))) → Não balanceados!

( ) → Balanceados!

Comandos de Repetição

DCC 120

Comandos iterativos em C

- `while` (enquanto..faça)
- `do...while` (faça..enquanto)
- `for` (para..faça)

while

- Sintaxe

```
while (condicao)
{
    blocoDeComandos;
 }
```

- Exemplo

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

do while

- Sintaxe

```
do
{
    blocoDeComandos;
} while (condicao);
```

- Exemplo

```
int main()
{
  int i;
  i=0;
  do
  {
    printf("%d\n",i);
    i++;
  } while (i<=10);
  return 0;
}
```

for

- Sintaxe

```
for ( inicializacao ; icondicao ; atualizacao )
{
  blocoDeComandos;
}
```

- Exemplo

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

Exercícios

1) Chico tem 1,50 metro e cresce 2 centímetros por ano, enquanto Zé tem 1,40 metro e cresce 3 centímetros por ano. Construa um programa que calcule e imprima quantos anos serão necessários para que Zé seja maior que Chico. Use o comando do-while.

2) Escrever um função que lê um valor N inteiro e positivo e que calcula e escreve o valor de E. Faça um programa para testar sua função.

$$E = 1 + 1/2 + 1/3 + ... + 1/N$$

3) Escrever uma função que recebe por parâmetro um valor N inteiro e positivo e que calcula e escreve o valor de E. Faça um programa para testar sua função.

$$E = 1 + 1/1! + 1/2! + 1/3! + ... + 1/N!$$

4) Escreva um programa que leia 10 valores (usando a mesma variável) e encontre o maior e o menor deles. Mostre o resultado.

5) Escreva um programa que leia 10 valores (usando a mesma variável) e imprima a posição do maior e a posição do menor deles na sequência. Por exemplo:

```
Digite 10 valores:
3 -2  9  2  7 -5  8  5  2  3
O 3o elemento eh o maior.
O 6o elemento eh o menor.
```

Exercícios

6) Faça um programa que, dado um conjunto de valores inteiros (fornecidos um a um pelo usuário), determine qual o menor valor do conjunto. O final do conjunto de valores é conhecido através do valor zero, que não deve ser considerado.

7) Usando uma função que converta graus Fahrenheit em Celsius (fórmula C=5*(F – 32)/9), escreva um programa que construa uma tabela de graus Celsius em função de Fahrenheit, de 50ºF a 150ºF, variando de 1 em 1. Use o comando for.

8) Faça uma função que receba um valor N como parâmetro e calcule e retorne o enésimo termo da série de Fibonacci. A série de Fibonacci é dada por: f1 = f2 = 1 f5 = f3 + f4 = 5
f3 = f1 + f2 = 2 f6 = f4 + f5 = 8
f4 = f2 + f3 = 3 ....

Faça um programa que imprima o 5º, o 10º, o 15º, o 20º e o 25º termos da série de Fibonacci, chamando a função acima para calcular cada termo.

Exercícios

DESAFIO: Neste exercício, você vai fazer um programa que retrata um jogo de adivinhação. O jogo funciona da seguinte forma:

- O programa avisa ao usuário que ele deve pensar em um número de 1 a 100
- O programa faz uma sequência de perguntas para o usuário. As perguntas devem ser do tipo:
  - O número é maior que 10 e menor ou igual a 20?
  - O número é 83?
- O usuário só pode responder ‘S’ (sim) ou ‘N’ (não).

O grande desafio é fazer um programa cuja estratégia permita sempre adivinhar o número fazendo até 7 perguntas.