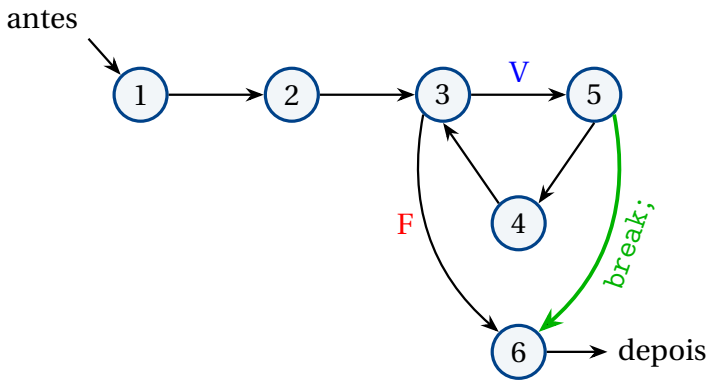
Figura 3.1: Fluxo de execução da estrutura de repetição *for*




**Fonte:** Elaborada pelo autor

### 3.1.4 Exercícios

**Exercício 3.1:** Escreva um programa que imprima os números inteiros de 0, inclusive, a 20, inclusive, usando a estrutura de repetição `for`.

**Arquivo com a solução: ex3.1.c**

**Saída**

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

**Exercício 3.2:** Escreva um programa que imprima os números inteiros pares que estão no intervalo entre 0, inclusive, e 50, inclusive, usando a estrutura de repetição `for`.

**Arquivo com a solução: ex3.2.c**

**Saída**

```
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44
46 48 50
```

**Exercício 3.3:** Escreva um programa que imprima os números inteiros de 20, inclusive, a 0, inclusive, usando a estrutura de repetição `for`

**Arquivo com a solução: ex3.3.c**

**Saída**

```
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
```

**Exercício 3.4:** Escreva um programa que apresente o quadrado dos números inteiros de 15, inclusive, a 30, inclusive, usando a estrutura de repetição `for`

**Arquivo com a solução: ex3.4.c**

**Saída**

```
225 256 289 324 361 400 441 484 529 576 625 676 729 784 841 900
```

**Exercício 3.5:** Escreva um programa que peça para o usuário entrar com um número inteiro maior ou igual a 0. Se um valor incorreto for digitado, o programa deve avisar o usuário imprimindo na tela a mensagem "Valor incorreto (negativo)" e terminar. Caso o número seja correto, o programa deve exibir os números de 0 ao número digitado.

**Arquivo com a solução: ex3.5.c**

**Entrada**

```
Forneca um numero maior ou igual a zero: 7
```

**Saída**

```
0 1 2 3 4 5 6 7
```

**Entrada**

```
Forneca um numero maior ou igual a zero: -5
```

**Saída**

```
Valor incorreto (negativo)
```

**Exercício 3.6:** Escreva um programa que peça para o usuário entrar com um número inteiro maior ou igual a 0. Se um valor incorreto for digitado, o programa deve avisar o usuário imprimindo na tela a mensagem "Valor incorreto (negativo)" e terminar. Caso o número seja correto, o programa deve exibir os números do número digitado até 0.

**Arquivo com a solução: ex3.6.c**

**Entrada**

```
Forneca um numero maior ou igual a zero: 7
```

**Saída**

```
7 6 5 4 3 2 1 0
```

**Entrada**

```
Forneca um numero maior ou igual a zero: -10
```

**Saída**

```
Valor incorreto (negativo)
```

**Exercício 3.7:** Escreva um programa que peça para o usuário entrar com um número inteiro menor ou igual a 0. Se um valor incorreto for digitado, o programa deve avisar o usuário imprimindo na tela a mensagem "Valor incorreto (positivo)" e terminar. Caso o número seja correto, o programa deve exibir os números do número digitado até 0.

**Arquivo com a solução: ex3.7.c**

**Entrada**

```
Forneca um numero menor ou igual a zero: -10
```

**Saída**

```
-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
```

**Entrada**

```
Forneca um numero menor ou igual a zero: 20
```

**Saída**

```
Valor incorreto (positivo)
```

**Exercício 3.8:** Escreva um programa que peça para o usuário entrar com um número inteiro menor ou igual a 0. Se um valor incorreto for digitado, o programa deve avisar o

usuário imprimindo na tela a mensagem “Valor incorreto (positivo)” e terminar. Caso o número seja correto, o programa deve exibir os números de 0 ao número digitado.

**Arquivo com a solução: ex3.8.c**

**Entrada**

```
Forneca um numero menor ou igual a zero: -9
```

**Saída**

```
0 -1 -2 -3 -4 -5 -6 -7 -8 -9
```

**Entrada**

```
Forneca um numero menor ou igual a zero: 15
```

**Saída**

```
Valor incorreto (positivo)
```

**Exercício 3.9:** Escreva um programa que peça para o usuário fornecer um número inteiro. O programa deve exibir a tabuada de 0 a 10 desse número.

**Arquivo com a solução: ex3.9.c**

**Entrada**

```
Tabuada do Numero: 5
```

**Saída**

```
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

**Exercício 3.10:** Escreva um programa que apresente se cada número inteiro no intervalo entre 45, inclusive, e 60, inclusive, é divisível ou não por 4. Utilize a estrutura de repetição `for`.

**Arquivo com a solução: ex3.10.c**

**Saída**

```
45: indivisivel
46: indivisivel
47: indivisivel
48: divisivel
49: indivisivel
50: indivisivel
51: indivisivel
52: divisivel
53: indivisivel
54: indivisivel
55: indivisivel
56: divisivel
57: indivisivel
58: indivisivel
59: indivisivel
60: divisivel
```

**Exercício 3.11:** Escreva um programa que peça para o usuário fornecer dois números inteiros. Se o primeiro número for menor ou igual ao segundo, o programa deve exibir todos os números no intervalo entre os números digitados em ordem crescente. Caso o primeiro número seja maior que o segundo, o programa deve exibir todos os números no intervalo entre os números digitados em ordem decrescente.

**Arquivo com a solução: ex3.11.c**

**Entrada**

```
N1: 2
N2: 10
```

**Saída**

```
2 3 4 5 6 7 8 9 10
```

**Entrada**

```
N1: 30
N2: 20
```

**Saída**

```
30 29 28 27 26 25 24 23 22 21 20
```

**Exercício 3.12:** Escreva um programa que peça para o usuário fornecer dois números inteiros. O programa deve contar e exibir a quantidade de números pares que existem no intervalo compreendido entre os dois números informados, considerando esses dois números. Fique atento à ordem de entrada dos números.

**Arquivo com a solução: ex3.12.c**

**Entrada**

```
N1: 5
N2: 100
```

**Saída**

```
Numeros pares entre 5 e 100: 48
```

**Entrada**

```
N1: 20
N2: -30
```

**Saída**

```
Numeros pares entre -30 e 20: 26
```

**Exercício 3.13:** Escreva um programa que conte quantos números inteiros múltiplos de 2, múltiplos de 3 e múltiplos de 4 existem no intervalo de dois números inteiros fornecidos pelo usuário. Esses contadores devem ser exibidos no final. Fique atento à ordem de entrada dos números.

**Arquivo com a solução: ex3.13.c**

**Entrada**

```
N1: -50
N2: 200
```

**Saída**

```
Multiplos de 2: 126
Multiplos de 3: 83
Multiplos de 4: 63
```

**Entrada**

```
N1: 50
N2: -100
```

**Saída**

```
Multiplos de 2: 76
Multiplos de 3: 50
Multiplos de 4: 38
```

**Exercício 3.14:** Escreva um programa que calcule o somatório dos valores compreendidos entre dois números inteiros fornecidos pelo usuário, incluindo tais números no cálculo. Fique atento à ordem de entrada dos números.

**Arquivo com a solução: ex3.14.c**

**Entrada**

```
N1: -5
N2: 50
```

**Saída**

```
Somatorio entre -5 e 50: 1260
```

**Entrada**

```
N1: 80
N2: -10
```

**Saída**

```
Somatorio entre -10 e 80: 3185
```

**Exercício 3.15:** Escreva um programa que peça para o usuário fornecer um número inteiro positivo e que calcule o fatorial desse número. Caso o número seja negativo, o programa deve avisar o usuário, usando a mensagem "Nao ha fatorial de numero negativo." (sem acentos) e terminar. Caso contrário o programa deve calcular o fatorial do número digitado e exibi-lo. Lembrando que:

- $n! = \prod_{i=1}^{n} i \ , \forall n \in \mathbb{N}^*$

**Arquivo com a solução: ex3.15.c**

**Entrada**

```
Numero: 5
```

**Saída**

```
5! = 120
```

**Entrada**

```
Numero: -10
```

**Saída**

```
Nao ha fatorial de numero negativo.
```

**Exercício 3.16:** Escreva um programa que exiba os vinte primeiros termos da série de Fibonacci. A série de Fibonacci inicia com 1 e 1, sendo os próximos termos gerados pela soma dos dois últimos termos. Os nove primeiros termos da série de Fibonacci são: 1 1 2 3 5 8 13 21 34. Obs: O primeiro termo é o termo 0, o segundo termo é o termo 1, o terceiro termo é o termo 2 e assim por diante.

**Arquivo com a solução: ex3.16.c**

**Saída**

```
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
6765
```

**Exercício 3.17:** Escreva um programa que peça ao usuário o termo da série de Fibonnaci que ele quer que seja obtido e que então exiba esse termo.

**Arquivo com a solução: ex3.17.c**

**Entrada**

```
Termo desejado: 0
```

**Saída**

```
Fibonacci de 0 e 1
```

**Entrada**

```
Termo desejado: 1
```

**Saída**

```
Fibonacci de 1 e 1
```

**Entrada**

```
Termo desejado: 5
```

**Saída**

```
Fibonacci de 5 e 8
```

**Entrada**

```
Termo desejado: 15
```

**Saída**

```
Fibonacci de 15 e 987
```

**Exercício 3.18:** Escreva um programa que exiba o seguinte desenho usando, obrigatoriamente, a estrutura de repetição `for`.

**Arquivo com a solução: ex3.18.c**

**Saída**

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

**Exercício 3.19:** Escreva um programa que exiba o seguinte desenho usando, obrigatoriamente, a estrutura de repetição `for`.

**Arquivo com a solução: ex3.19.c**

**Saída**

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

**Exercício 3.20:** Escreva um programa que exiba o seguinte desenho usando, obrigatoriamente, a estrutura de repetição `for`. Os asteriscos quase pretos indicam espaços.

**Arquivo com a solução: ex3.20.c**

**Saída**

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

*****
****
***
**
*

****
***
**
*
*****

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

**Exercício 3.21:** Escreva um programa que leia uma altura em inteiro e imprima um triângulo de asteriscos, baseado nessa altura. Uma altura negativa deve resultar em um triângulo de cabeça para baixo. Uma altura igual a zero não produzirá um triângulo. Os asteriscos quase pretos indicam espaços.

**Arquivo com a solução: ex3.21.c**

**Entrada**

```
Altura: 3
```

**Saída**

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

**Entrada**

```
Altura: 11
```

**Saída**

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

**Entrada**

```
Altura: -5
```

**Saída**

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

**Exercício 3.22:** Escreva um programa que peça para o usuário fornecer 5 números positivos. Caso algum seja menor ou igual a zero, após o usuário fornecer todos os números, o programa deve avisar o usuário e terminar com a mensagem "Forneca apenas numeros positivos.". Os valores fornecidos devem ser usados para desenhar um gráfico de barras usando o símbolo asterisco. Os asteriscos quase pretos indicam espaços.

**Dica:** Para formatar um valor inteiro usando zeros à esquerda para preenchimento, use o especificador de formato `%0nd`, onde `n` é a quantidade de casas que o valor e os zeros ocuparão. Por exemplo, uma variável que contenha o valor `15` ao ser formatada usando o especificador de formato `%04d` resultará em `0015`, ou seja, o número inteiro, precedido de zeros, ocupando quatro casas. Veja o código abaixo:

```c
int n = 15;
printf( "%04d", n );   // imprime 0015
```

**Arquivo com a solução: ex3.22.c**

**Entrada**

```
N1: 3
N2: 5
N3: 10
N4: 2
N5: 6
```

**Saída**

```
0010**********
0009*********
0008********
0007*******
0006******
0005*****
0004****
0003***
0002**
0001*
```

**Entrada**

```
N1: 4
N2: 8
N3: 0
N4: -7
N5: 2
```

**Saída**

```
Forneca apenas numeros positivos.
```

**Exercício 3.23:** Escreva um programa para ler as notas de 10 alunos de uma turma e calcular e exibir a média aritmética destas notas. Armazene a média em uma variável. As notas são números decimais. Utilize a estrutura de repetição `for` para coletar as

notas.

**Arquivo com a solução: ex3.23.c**

**Entrada**

```
Forneca a nota de 10 alunos:
Nota 01: 6
Nota 02: 8
Nota 03: 9
Nota 04: 8.75
Nota 05: 7
Nota 06: 5
Nota 07: 6
Nota 08: 7.5
Nota 09: 8
Nota 10: 9
```

**Saída**

```
A media aritmetica das dez notas e: 7.43
```

## 3.2 Estruturas de Repetição *while* e *do ... while*

### 3.2.1 Exemplos em Linguagem C

Estrutura do *while* - Verifique a Figura 3.2

```
1 // antes
2 (1)
3           (2)
4 while ( teste ) {
5     (3)
6 }
7 (4)
8 // depois
```

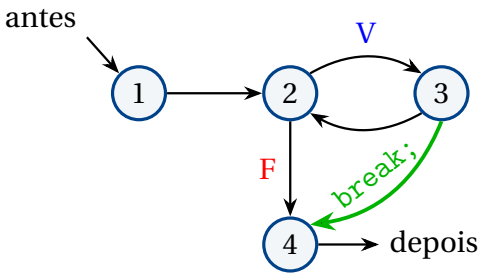
Estrutura do *do ... while* - Verifique a Figura 3.3

```
1 // antes
2 (1)
```

```
3 do {
4     (2)
5             (3)
6 } while ( teste );
7 (4)
8 // depois
```

### 3.2.2 Diagrama de Fluxo da Estrutura de Repetição *while*

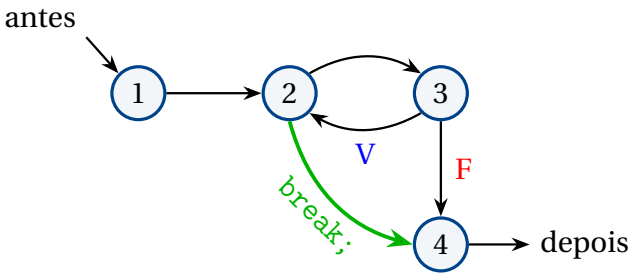
Figura 3.2: Fluxo de execução da estrutura de repetição *while*




**Fonte:** Elaborada pelo autor

### 3.2.3 Diagrama de Fluxo da Estrutura de Repetição *do ... while*

Figura 3.3: Fluxo de execução da estrutura de repetição *do ... while*




**Fonte:** Elaborada pelo autor

### 3.2.4 Exercícios

**Exercício 3.24:** Escreva um programa que solicite a idade de várias pessoas e imprima o total de pessoas com menos de 21 anos e o total de pessoas com mais de 50 anos. O programa deve terminar e exibir os resultados quando a idade fornecida for negativa.

**Arquivo com a solução: ex3.24.c**

**Entrada**

```
Idade da pessoa 01: 10
Idade da pessoa 02: 55
Idade da pessoa 03: -1
```

**Saída**

```
Total de pessoas menores de 21 anos: 1
Total de pessoas com mais de 50 anos: 1
```

**Entrada**

```
Idade da pessoa 01: 9
Idade da pessoa 02: 15
Idade da pessoa 03: 57
Idade da pessoa 04: 20
Idade da pessoa 05: 23
Idade da pessoa 06: 19
Idade da pessoa 07: 43
Idade da pessoa 08: 66
Idade da pessoa 09: -10
```

**Saída**

```
Total de pessoas menores de 21 anos: 4
Total de pessoas com mais de 50 anos: 2
```

**Exercício 3.25:** Escreva um programa que efetue a leitura sucessiva de valores numéricos decimais e apresente no final o somatório, a média e a quantidade de valores lidos, armazenada como inteiro. O programa deve continuar lendo os números até que seja fornecido um número negativo. Esse número negativo não deve entrar nos cálculos! Formate a saída dos números decimais usando 2 casas de precisão. Caso o número negativo seja o primeiro a ser fornecido, o programa deve exibir um somatório igual a zero, uma média igual a zero e uma quantidade igual a zero.

**Arquivo com a solução: ex3.25.c**

**Entrada**

```
Entre com um valor: 4
Entre com um valor: 8
Entre com um valor: -1
```

**Saída**

```
Somatorio: 12.00
Media: 6.00
Quantidade: 2
```

**Entrada**

```
Entre com um valor: 5
Entre com um valor: 8
Entre com um valor: 10
Entre com um valor: 15
Entre com um valor: 2
Entre com um valor: 9
Entre com um valor: 3
Entre com um valor: 2
Entre com um valor: -1
```

**Saída**

```
Somatorio: 54.00
Media: 6.75
Quantidade: 8
```

**Entrada**

```
Entre com um valor: -5
```

**Saída**

```
Somatorio: 0.00
Media: 0.00
Quantidade: 0
```

**Exercício 3.26:** Escreva um programa que efetue a leitura sucessiva de valores numé-

ricos inteiros e apresente no final o menor e o maior número que foram fornecidos. O programa deve continuar lendo os números até que seja fornecido um número negativo, que por sua vez não deve ser apresentado como menor ou maior número. Caso o número negativo seja o primeiro a ser fornecido, o programa deve exibir tanto o menor quanto o maior número como zero. Pense no que precisa ser feito para inicializar apropriadamente os valores das variáveis menor e maior.

**Arquivo com a solução: ex3.26.c**

**Entrada**

```
Entre com um valor: 7
Entre com um valor: 15
Entre com um valor: 3
Entre com um valor: 29
Entre com um valor: 2
Entre com um valor: 103
Entre com um valor: 0
Entre com um valor: 34
Entre com um valor: -1
```

**Saída**

```
Menor numero: 0
Maior numero: 103
```

**Entrada**

```
Entre com um valor: 5
Entre com um valor: -1
```

**Saída**

```
Menor numero: 5
Maior numero: 5
```

**Entrada**

```
Entre com um valor: -5
```

**Saída**

```
Menor numero: 0
Maior numero: 0
```

**Exercício 3.27:** Escreva um programa para ler um número indeterminado de dados de pesos de pessoas como números decimais. O último dado, que não entrará nos cálculos, deve ser um valor negativo. O programa deve calcular e imprimir a média aritmética dos pesos das pessoas que possuem mais de 60kg e o peso do indivíduo mais pesado. Formate a saída dos números decimais usando 2 casas de precisão. Caso o número negativo seja o primeiro a ser fornecido, o programa deve exibir a média e o peso mais pesado ambos como zero.

**Arquivo com a solução: ex3.27.c**

**Entrada**

```
Entre com o peso da pessoa 01: 55.8
Entre com o peso da pessoa 02: 102.7
Entre com o peso da pessoa 03: 86.3
Entre com o peso da pessoa 04: -1
```

**Saída**

```
Media dos pesos acima de 60kg: 94.50
A pessoa mais pesada possui 102.70kg
```

**Entrada**

```
Entre com o peso da pessoa 01: -1
```

**Saída**

```
Media dos pesos acima de 60kg: 0.00
A pessoa mais pesada possui 0.00kg
```

**Entrada**

```
Entre com o peso da pessoa 01: 30.0
Entre com o peso da pessoa 02: -1
```

**Saída**

```
Media dos pesos acima de 60kg: 0.00
A pessoa mais pesada possui 30.00kg
```

**Entrada**

```
Entre com o peso da pessoa 01: 90.0
Entre com o peso da pessoa 02: -1
```

**Saída**

```
Media dos pesos acima de 60kg: 90.00
A pessoa mais pesada possui 90.00kg
```

**Exercício 3.28:** Escreva um programa para ler o saldo inicial de uma conta bancária, um valor decimal. A seguir ler um número indeterminado de pares de valores indicando respectivamente o tipo da operação (codificado da seguinte forma: 1.Depósito 2.Retirada e 3.Fim) e o valor que será movimentado. Quando for informado para o tipo da operação o código 3, o programa deve ser encerrado e impresso o saldo final da conta com as seguintes mensagens: “Sem saldo.” caso o saldo seja zero, “Conta devedora.”, se o saldo for negativo ou “Conta preferencial.”, se o saldo seja positivo. Caso seja fornecido um tipo incorreto de operação, ou seja, diferente de 1, 2 ou 3, o programa deve exibir ao usuário a mensagem “Operacao invalida.” e solicitar novamente a operação. Formate a saída dos números decimais usando 2 casas de precisão.

**Arquivo com a solução: ex3.28.c**

**Entrada**

```
Saldo inicial: 3000
Operacoes:
    1) Deposito;
    2) Saque;
    3) Fim.
Operacao desejada: 1
Valor a depositar: 500
Operacao desejada: 1
Valor a depositar: 300
Operacao desejada: 1
Valor a depositar: 100
Operacao desejada: 2
Valor a sacar: 2555
Operacao desejada: 3
```

**Saída**

```
Saldo final: R$1345.00
Conta preferencial.
```

**Entrada**

```
Saldo inicial: 1000
Operacoes:
    1) Deposito;
    2) Saque;
    3) Fim.
Operacao desejada: 2
Valor a sacar: 500
Operacao desejada: 2
Valor a sacar: 300
Operacao desejada: 2
Valor a sacar: 300
Operacao desejada: 3
```

**Saída**

```
Saldo final: -R$100.00
Conta devedora.
```

**Entrada**

```
Saldo inicial: 2000
Operacoes:
    1) Deposito;
    2) Saque;
    3) Fim.
Operacao desejada: 2
Valor a sacar: 1500
Operacao desejada: 2
Valor a sacar: 500
Operacao desejada: 3
```

**Saída**

```
Saldo final: R$0.00
Sem saldo.
```

**Exercício 3.29:** Escreva um programa para ler 2 valores inteiros e imprimir o resultado da divisão do primeiro pelo segundo. Se o segundo valor informado for zero, deve ser impressa uma mensagem de "Nao existe divisao inteira por zero!" (sem acentos) e lido um novo valor. Ao final do programa, deve ser impressa a seguinte mensagem: "Voce deseja realizar outro calculo? (S/N): " Se a resposta for 'S' o programa deverá retornar ao começo, repetindo o processo, caso contrário deverá encerrar a sua execução imprimindo quantos cálculos foram feitos.

**Arquivo com a solução: ex3.29.c**

**Entrada**

```
N1: 10
N2: 5
Voce deseja realizar outro calculo? (S/N): S
N1: 11
N2: 3
Voce deseja realizar outro calculo? (S/N): S
N1: 15
N2: 0
Entre novamente com N2: 0
Entre novamente com N2: 5
Voce deseja realizar outro calculo? (S/N): N
```

**Saída**

```
10 / 5 = 2
11 / 3 = 3
Nao existe divisao inteira por zero!
Nao existe divisao inteira por zero!
15 / 5 = 3
```