

Comandos de repetição

- Repetição determinística

`for`

- Repetição não determinística

`while`

`do-while`

Comando while

- Repetição condicional
- **testa a condição no início**, e só executa se a condição for verdadeira

```
while (condição) {  
    comandos;  
}
```

- onde:
 - condição: é uma expressão lógica (retorna V ou F)

for x while

```
for (inicialização; condição; atualização) {  
    comandos;  
}
```

```
inicialização;  
while (condição) {  
    comandos;  
    atualização;  
}
```

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (1)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

Não considerado
no cálculo



Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (2)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (3)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (4)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (5)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (6)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (7)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (8)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V
12	2	4	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (9)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V
12	2	4	---
16	0	4	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (10)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V
12	2	4	---
16	0	4	---
10	0	4	F

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (11)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V
12	2	4	---
16	0	4	---
10	0	4	F
18	0	4	---

Ex.1: lê sequência de inteiros, calculando e imprimindo seu quadrado. **Marca de parada:** leitura do valor 0 (12)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

linha	num	quad	teste
4	?	?	---
8	5	?	---
10	5	?	V
12	5	25	---
16	2	25	---
10	2	25	V
12	2	4	---
16	0	4	---
10	0	4	F
18	0	4	---

Repetição com teste no início – quesitos importantes em repetições

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, quad;
5
6     // imprime uma msg e lê o 1o. inteiro
7     printf("Digite um número inteiro: ");
8     scanf("%d", &num);
9
10    while( num != 0 )
11    {
12        quad = num * num;
13        printf("\nQuadrado de %d: %d", num, quad);
14
15        printf("\nDigite um número inteiro: ");
16        scanf("%d", &num);
17    }
18    return 0;
19 }
```

Inicialização

Toda variável da condição precisa ser inicializada antes do laço, através de atribuição ou leitura.

Condição

Teste normalmente envolve ao menos uma variável.

Atualização

Ao menos uma variável da condição precisa ser atualizada no interior do laço.

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada: leitura do valor 0 (1)**

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

Usa acumulador

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (2)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (3)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (4)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

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

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (5)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

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

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (6)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

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

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (7)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (8)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	
8	-2	9	V

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (9)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	
8	-2	9	V
10	-2	7	

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (10)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	
8	-2	9	V
10	-2	7	
14	0	7	

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (11)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	
8	-2	9	V
10	-2	7	
14	0	7	
8	0	7	F

Ex.2: lê sequência de inteiros, calculando e imprimindo a sua soma. **Marca de parada:** leitura do valor 0 (12)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

linha	num	soma	teste
4	?	?	
5	?	0	
7	9	0	
8	9	0	V
10	9	9	
14	-2	9	
8	-2	9	V
10	-2	7	
14	0	7	
8	0	7	F
16	0	7	

Acumuladores precisam de:

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um número inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

Inicialização

Variável precisa ser inicializada (normalmente com zero)

Atualização

Variável precisa conservar o valor acumulado e acrescentar novo valor

Impressão de acumuladores

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, soma;
5     soma = 0; // inicializa acumulador
6     printf("Digite um número inteiro: ");
7     scanf("%d", &num);
8     while( num != 0 )
9     {
10         soma = soma + num; //atualiza acumulador
11         printf("\nSoma parcial: %d",soma);
12         printf("\nDigite um numero inteiro: ");
13         scanf("%d", &num);
14     }
15     printf("\nSoma total: %d",soma);
16     return 0;
17 }
18 }
```

Valor parcial

Impressão da variável é feita no laço, após sua atualização

Valor total

Impressão da variável é feita após o laço

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (1)

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

Usa contador

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (2)

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

linha	num	cont	teste
4	?	?	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (3)

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

linha	num	cont	teste
4	?	?	
5	?	0	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (4)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (5)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (6)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (7)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (8)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	
8	13	1	V

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (9)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	
8	13	1	V
10	13	2	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (10)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	
8	13	1	V
10	13	2	
12	0	2	

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (11)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	
8	13	1	V
10	13	2	
12	0	2	
8	0	2	F

Ex.3: lê sequência de inteiros e imprime a quantidade de valores lidos. **Marca de parada:** leitura do valor 0 (12)

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

linha	num	cont	teste
4	?	?	
5	?	0	
7	22	0	
8	22	0	V
10	22	1	
12	13	1	
8	13	1	V
10	13	2	
12	0	2	
8	0	2	F
14	0	2	

Contadores precisam de:

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

Inicialização

Variável precisa ser inicializada (normalmente com zero)

Atualização

Variável precisa ser atualizada (incrementada)

Exercícios com o comando while

- 1) Faça um programa que leia vários números inteiros, verifique e imprima a quantidade de primos dentre os números digitados. O programa acaba quando for digitado um valor menor ou igual a zero.
 - 2) Faça um programa que leia um número indeterminado de duplas **m** e **n**, todos inteiros e positivos, uma dupla de cada vez. Para cada dupla lida, o programa deve chamar uma função que receba **m** e **n** e calcule a soma dos **n** inteiros consecutivos a partir de **m** inclusive. Ao final, a função deve retornar o valor da soma para a função main, que irá imprimi-la. (ex.: para $m = 2$ e $n = 4$, a função irá calcular $2+3+4+5$)
-

Exercícios com o comando while

- 3) Faça um programa que leia um número inteiro e positivo. A seguir, chame uma função que receba esse número e verifique se o mesmo é triangular, retornando essa informação à função main, que irá imprimir uma mensagem adequada. Um número é dito triangular quando é resultado do produto de 3 números consecutivos. (ex.: 24 é triangular, pois $24 = 2 \cdot 3 \cdot 4$)
 - 4) Chico tem 1,50m e cresce 2 centímetros por ano, enquanto Juca tem 1,10m e cresce 3 centímetros por ano. Faça um programa que calcule e imprima quantos anos serão necessários para que Juca seja maior que Chico.
-

Exercícios com o comando while

- 5) Faça um programa que leia um número inteiro N e chame uma função passando N como parâmetro. A função deve verificar se N é um quadrado perfeito, informando sua raiz, calculada através da soma de ímpares. Um número é um quadrado perfeito se for possível somarmos números ímpares consecutivos ($1+3+5+7+\dots$) até que esta soma seja igual ao próprio número. Neste caso, a quantidade de termos somados é a raiz quadrada deste número. (ex. $16 = 1+3+5+7$. Foram somados 4 termos, que é a raiz quadrada de 16).