

# Aula anterior: 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)
-

**Ex.1 (usando **while**):** lê um nro indeterminado de valores inteiros e verifica quantos são negativos. O programa encerra ao ser lido o valor 0

---

```
1  #include <stdio.h>
2
3  ▼ int main(){
4      int num, cont = 0;
5
6      printf("digite valor inteiro (0 para parar): \n");
7      scanf("%d", &num);
8
9      ▼ while (num != 0){
10         if (num < 0)
11             cont++;
12         printf("digite valor inteiro (0 para parar): \n");
13         scanf("%d", &num);
14     }
15     printf("quantidade de nros negativos = %d\n", cont);
16
17     return 0;
18 }
```

Testa condição  
no início

```
digite valor inteiro (0 para parar):
3
digite valor inteiro (0 para parar):
-7
digite valor inteiro (0 para parar):
1
digite valor inteiro (0 para parar):
0
quantidade de nros negativos = 1
```

# Comandos de repetição

---

- Repetição determinística

`for`

- Repetição não determinística

`while`

`do-while`

# Comando do while

---

- Repetição condicional
- **testa a condição no final**, e executa pelo menos uma vez

```
do {  
    comandos;  
} while condição;
```

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

**Ex.2 (usando **do while**):** lê um nro indeterminado de valores inteiros e verifica quantos são negativos. O programa encerra ao ser lido o valor 0

---

```
1  #include <stdio.h>
2
3  ▼ int main(){
4      int num, cont = 0;
5
6      ▼ do{
7          printf("digite valor inteiro (0 para parar): \n");
8          scanf("%d", &num);
9          if (num < 0)
10             cont++;
11      } while(num != 0);
12      printf("quantidade de nros negativos = %d\n", cont);
13
14      return 0;
15  }
16
```

Testa condição  
no final  
→ executa, pelo  
menos, 1 vez

```
digite valor inteiro (0 para parar):
3
digite valor inteiro (0 para parar):
-7
digite valor inteiro (0 para parar):
1
digite valor inteiro (0 para parar):
0
quantidade de nros negativos = 1
```

# Comparação **while** x **do while**

```
1 #include <stdio.h>
2
3 int main(){
4     int num, cont = 0;
5
6     printf("digite valor inteiro (0 para parar): \n");
7     scanf("%d", &num);
8
9     while (num != 0){
10         if (num < 0)
11             cont++;
12         printf("digite valor inteiro (0 para parar): \n");
13         scanf("%d", &num);
14     }
15     printf("quantidade de nros negativos = %d\n", cont);
16
17     return 0;
18 }
```

```
1 #include <stdio.h>
2
3 int main(){
4     int num, cont = 0;
5
6     do{
7         printf("digite valor inteiro (0 para parar): \n");
8         scanf("%d", &num);
9         if (num < 0)
10             cont++;
11     } while(num != 0);
12     printf("quantidade de nros negativos = %d\n", cont);
13
14     return 0;
15 }
```

# Término laço

---

- Controle do término de um laço (nro de repetições indeterminado)
  - através de uma das variáveis de entrada de dados, informando um valor fora do intervalo de valores válidos
    - Marca de parada
  - perguntar explicitamente p/ o usuário, ao final de cada iteração, se ele quer continuar ou parar
    - interessante quando os valores de entrada podem assumir qualquer valor (positivo, negativo, etc.)
    - exige uma entrada a mais p/ cada conjunto de dados

## **Exercícios (do while): Faça um programa que ...**

---

- 1) Leia uma quantidade não determinada de valores positivos, calcule e imprima a soma dos mesmos. Usar um valor negativo como marca de parada.
  
- 2) Leia uma quantidade não determinada de valores positivos, calcule e imprima o produto dos mesmos e a quantidade de valores multiplicados. Usar um valor negativo menor ou igual a zero como marca de parada.



## **Exercícios (do while): Faça um programa que ...**

---

3) Calcule o MDC entre 2 números inteiros lidos do usuário.

4) Implemente um jogo de adivinhar. O programa deve gerar um número aleatório (entre 0 e 9, inclusos), o qual deve ser adivinhado pelo usuário. Quando o usuário conseguir adivinhar, o programa deve exibir uma mensagem informando o acerto e o número de tentativas necessárias. A seguir, o programa deve perguntar se o usuário deseja jogar novamente e, em caso positivo, possibilitar a repetição do jogo.

---

## Exercícios (do-while): Faça um programa que ...

---

- 5) Calcule o menor inteiro positivo cujo quadrado é superior a um número inteiro e positivo lido do usuário.
  
- 6) Calcule a média dos números pares digitados pelos usuário. O programa deve encerrar a entrada quando encontrar o valor 0.
  - a) faça uma versão com while
  - b) faça uma versão com do-while