

## Introdução breve

Como vimos na aula há diversos operadores em C:

- aritméticos: soma (+), subtração (-), multiplicação (\*) e divisão (/)
- relacionais: maior (>), maior ou igual (>=), menor (<), menor ou igual (<=), igual (==) e diferente (!=)
- lógicos: e (&&), ou (||) e negação (!), em inglês and, or e not respectivamente
- de atribuição: atribuição (=)

Tanto os operadores aritméticos como o de atribuição já usamos bastante e vimos que no caso dos operadores aritméticos eles seguem a mesma regra de precedência que estamos acostumados a utilizar no nosso dia a dia. Os outros operadores utilizaremos mais à frente no nosso curso.

## Saída Padrão

A saída padrão é o console. Para imprimirmos ou exibirmos algo no console, utilizamos a função **printf** do C:



```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World");
5
6      return 0;
7  }
8

input
Hello World

...Program finished with exit code 0
Press ENTER to exit console.
```

No Código acima, na linha 4, colocamos a na função **printf** o conteúdo que está em formato tipo texto (pois é um conjunto de caracteres entre aspas duplas – "). Como vemos na imagem abaixo, após executar nosso programa, foi impresso no console o que estava dentro das aspas, ou seja, o conteúdo do tipo texto.

Para imprimirmos conteúdo tipo numérico, como números inteiro ou reais temos de utilizar um parâmetro dentro da string que dirá a função quem queremos imprimir um número, e após a string, colocamos uma vírgula e o número a imprimir. No caso de impressão de números inteiros, esse parâmetro é o `%d` e no caso de números reais, `%f`.

```
main.c
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello World, %d",3);
5
6     return 0;
7 }
8
```

Hello World, 3

...Program finished with exit code 0  
Press ENTER to exit console.

No código acima, colocamos o parâmetro %d e depois o número inteiro 3.

```
main.c
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello World, %f",3.1428);
5
6     return 0;
7 }
8
```

Hello World, 3.142800

...Program finished with exit code 0  
Press ENTER to exit console.

No código acima, temos que foi impresso o número real 3.1428, e para sua impressão usamos o f e não o d, pois o f é utilizado para número desse tipo. Note que além dos quatro decimais, foram impressos mais dois 0, isso porque por padrão todo número decimal sempre é impresso com seis casas decimais. Podemos ajustar o número de casas decimais, colocando um ponto seguido do número de casas decimais que queremos imprimir entre o % e o f, como ilustrado na figura com código a seguir.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World, %.4f",3.1428);
5
6      return 0;
7  }
8
```

Hello World, 3.1428

...Program finished with exit code 0  
Press ENTER to exit console.

Fique atento pois mesmo que você defina um decimal com mais casas do que as que você delimitou na impressão, só será impresso o número que você determinou, como podemos observar no código abaixo.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World, %.2f",3.1428);
5
6      return 0;
7  }
8
```

Hello World, 3.14

...Program finished with exit code 0  
Press ENTER to exit console.

O tipo texto também tem um parâmetro de impressão, o %s. No código a seguir, podemos ver um exemplo do uso desse parâmetro.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World, %s", "meu texto");
5
6      return 0;
7  }
8
```

▼ ↗ 📄

```
Hello World, meu texto

...Program finished with exit code 0
Press ENTER to exit console.
```

De modo similar o tipo caractere que é definido entre aspas simples também tem seu parâmetro para impressão, o **%c**. Como observado no exemplo abaixo.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World, %c", 'e');
5
6      return 0;
7  }
8
```

▼ ↗ 📄

```
Hello World, e

...Program finished with exit code 0
Press ENTER to exit console.
```

Se quisermos imprimir mais de um número, texto ou caractere, basta adicionar os parâmetros correspondentes, e depois da string de impressão, os valores que queremos imprimir. A figura a seguir ilustra a impressão de quatro valores, cada um de tipo diferente, mas todos poderiam ser do mesmo tipo. A quantidade e o tipo é você que define.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World, %c - %s - %f - %d", 'e', "texto", 3.1428, 7);
5
6      return 0;
7  }
8
```

input

Hello World, e - texto - 3.142800 - 7

...Program finished with exit code 0  
Press ENTER to exit console.

**OBSERVAÇÃO:** Tome cuidado, use o parâmetro correto para cada tipo, senão sua impressão pode gerar algo que você não estava esperando ou a depender do compilador, pode gerar erro.

Note que utilizamos “-” para separar nossos valores dentro da string de impressão, mas você pode usar qualquer coisa, como espaços em branco, vírgulas, parênteses, fica a seu critério.

Além de valores também podemos imprimir variáveis, seja tipo texto, caractere ou número. Para isto basta colocar após a string as variáveis que queremos imprimir. Só esteja atento para utilizar para cada tipo o parâmetro correto.

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

Hello World, 7

...Program finished with exit code 0  
Press ENTER to exit console.

No código acima, criamos uma variável tipo **int** (inteira) e depois imprimimos seu valor na saída padrão.

O número e o tipo de variáveis a imprimir vai depender da sua necessidade, mas como no exemplo, é só colocar o parâmetro adequado ao tipo e depois a variável, e no caso de mais de uma variável, como anteriormente, separando-as por vírgulas, como no exemplo abaixo.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int a = 7;
5      float b = 3.128;
6      printf("Hello World, %d ,|%.3f ", a, b);
7
8      return 0;
9  }
```

inp

Hello World, 7 , 3.128

...Program finished with exit code 0  
Press ENTER to exit console.

## Entrada padrão

A entrada padrão no computador é o teclado. É a partir dele que temos interação com o usuário do nosso programa.

Para ler um valor da entrada padrão, utilizamos a função **scanf** do C. Para utilizarmos o valor que lemos da entrada, também temos de criar uma variável para armazená-lo.

O **scanf** é bem parecido com o **printf**, utilizamos uma string e um parâmetro que diz ao compilador, qual o tipo que queremos ler. Esses parâmetros são os mesmos do **printf**:

- **%d** – para lermos um número inteiro
- **%f** – para lermos um número real
- **%c** – para lermos um caractere
- **%s** – para lermos um texto

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int a;
5      scanf("%d", &a);
6
7      float b;
8      scanf("%f", &b);
9
10     char c;
11     scanf("%c", &c);
12
13     return 0;
14 }
15
```

No código acima criamos três variáveis, a, b e c dos tipos inteiro, real e caractere, respectivamente, e depois fazemos a três leituras e armazenamos os valores lidos em cada uma das variáveis.

Ao executar um código de leitura, é provável que o console mostre o cursor piscando, isso significa que o console está esperando uma entrada do usuário e essa espera só acaba quando o usuário aperta Enter, que é um sinal de que o usuário terminou de digitar a entrada. Se você coloca três **scanf**s o computador espera que você aperte Enter três vezes, um para cada operação de leitura. Assim, para cada operação de leitura, é aguardado uma confirmação com Enter. Se há uma leitura, o computador fica parado, esperando a confirmação e só executa o resto do seu código após o Enter que indica que a leitura foi encerrada.

Note que há antes do nome da variável, um “&”. Isso é um operador de endereço e se não for colocado antes do nome da variável no scanf, o valor não será armazenado na variável.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int a;
5      scanf("%d", &a);
6
7      printf("meu valor é: %d",a) ;
8
9      return 0;
10 }
11
```

```
7
meu valor é: 7

...Program finished with exit code 0
Press ENTER to exit console.
```

No exemplo acima, vemos um exemplo de programa em que é lido um valor inteiro e em seguida esse valor é impresso na saída padrão.

Além de imprimir, podemos fazer operações com o valor lido, manipulando esse valor do modo que desejamos.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int a;
5      scanf("%d", &a);
6
7      printf("meu valor é: %d",a) ;
8
9      a = a +1;
10     printf(", meu valor agora é: %d",a) ;
11
12     return 0;
13 }
14
```

```
7
meu valor é: 7, meu valor agora é: 8

...Program finished with exit code 0
Press ENTER to exit console.
```

No exemplo acima o usuário digitou 7, que foi armazenado na variável **a** cujo valor foi impresso, como observamos no código da linha 7. Em seguida, o valor de **a** foi incrementado de 1, ou seja, foi adicionado 1 ao valor de **a**, o que significa que o novo valor de **a** é 8. O que está de acordo com a saída do console.

Dica: se quisermos que a string de impressão contenha quebra de linha (Enter), basta adicionar o caractere “\n” dentro da string. A quantidade de “\n” fica a seu critério. Na figura a seguir vemos um exemplo de uso do “\n”, ou quebra de linha.

```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int a;
5      scanf("%d", &a);
6
7      printf("meu valor é: %d \n",a) ;
8
9      a = a +1;
10     printf("meu valor agora é: %d",a) ;
11
12     return 0;
13 }
14
```

```
7
meu valor é: 7
meu valor agora é: 8

...Program finished with exit code 0
Press ENTER to exit console.
```

**Nota: por enquanto não faremos a leitura de tipo texto.**

### Exercícios

Agora que já sabemos imprimir valores, inclusive os valores que estão em variáveis; além de ler da entrada padrão, vamos fazer alguns exercícios.

1. Crie um programa que imprima seu nome no console.
2. Crie um programa que imprima os dois últimos dígitos do seu CPF.
3. Crie um programa que leia um valor decimal (real) e o imprima na tela.
4. Com o código da questão anterior, faça com que na sua impressão só apareça um dígito depois do ponto.
5. Crie um programa que leia um número e imprima o dobro do número.
6. Crie um programa que leia dois números e em seguida imprima a soma dos dois.
7. Crie um programa que leia dois números e em seguida imprima a divisão do primeiro pelo segundo número lido.
8. (Desafio) Crie um programa que imprima o desenho:

```
|-|
|--|
|---|
|----|
|-----|
```

Do material do Marlan:

1. Faça um programa para ler dois valores e imprimir a soma dos mesmos.
2. Faça um programa para ler dois valores e imprimir a multiplicação dos mesmos.
3. Faça um programa para ler três valores e imprimir a soma dos mesmos.
4. Faça um programa para ter um valor em quilogramas (Kg) e imprimir o equivalente em gramas (g).
5. Faça um programa para ler um valor e imprimir o quadrado deste número.



6. Faça um programa para ler um valor correspondente a um lado de um quadrado e mostrar a **ÁREA** (usar float).
7. Faça um programa para ler um valor em polegadas e escrever o equivalente em centímetros. 1 polegada = 2.54 centímetros
8. Escreva um programa para ler a idade de uma pessoa e exibir quantos dias de vida ela possui. Considere sempre anos "cheios" e que um ano possui 365 dias.
9. Faça um programa para ler dois valores (variáveis A e B) e efetuar as operações de adição, subtração, multiplicação e divisão de A por B apresentado os quatro resultados obtidos.
10. Faça um programa que leia as 4 notas de um aluno e a seguir calcule a média.