

# Primeiros Passos

## Professor Wladimir

### Primeiros Passos

A estrutura básica de um programa em linguagem C é a seguinte:

```
1 #include <stdio.h>
2 int main() {
3 }
```

Na linha 1, utilizamos um comando para incluir a biblioteca de funções `stdio.h`<sup>1</sup>. Essa biblioteca fornece funções para realizar operações de leitura (entrada) e escrita (saída).

```
1 #include <stdio.h>
2 int main() {
3     printf("Hello World!!!");
4 }
```

Basicamente, a função `printf(texto)` recebe um texto e mostra no dispositivo de saída. A saída desse programa será:

```
1 Hello World!!!
```

### Imprimindo duas informações em linhas separadas

```
1 #include <stdio.h>
2 int main() {
3     printf("Hello World!!!");
4     printf("Hello Fundamentos de Programação!!!");
5 }
```

A saída desse programa será:

```
1 Hello World!!!Hello Fundamentos de Programação!!!
```

Para separar a saída em duas linhas, vamos utilizaremos um caractere especial `\n` para pular a linha.

```
1 #include <stdio.h>
2 int main() {
3     printf("Hello World!!!\n");
4     printf("Hello Fundamentos de Programação!!!\n");
5 }
```

A saída desse programa será:

```
1 Hello World!!!
2 Hello Fundamentos de Programação!!!
3
```

---

<sup>1</sup><https://cplusplus.com/reference/cstdio/>

## Variável inteira

Em um programa, utilizamos a memória do computador para armazenar valores que podem ser alterados durante a execução. Esses valores são representados por variáveis, que funcionam como espaços nomeados na memória. Esses espaços da memória teremos um tipo associado.

```
1 #include <stdio.h>
2 int main() {
3     int a;
4     a = 2; // a vale 2
5     a = a + 3; // a vale 5
6     a = a * 3; // a vale 15
7     a = a / 7; // a vale 2
8 }
```

Na linha 3, declaramos a variável `a` do tipo inteiro. Na linha 4, atribuímos a ela o valor 2. Ao longo do código, realizamos operações matemáticas sobre `a`, atualizando seu valor. Na linha 7, devido à divisão inteira, o valor resultante será 2, pois em C/C++ a divisão entre inteiros descarta a parte decimal.

O que você acha que vai acontecer nesse programa?

```
1 #include <stdio.h>
2 int main() {
3     int a;
4     a = 2; // a vale 2
5     a = a + 3; // a vale 5
6     a = a * 3; // a vale 15
7     a = a / 7.0; // a vale 2
8 }
```

A variável `a` continua valendo 2. Na linha 7, a linguagem C/C++ realiza a divisão real contudo a parte decimal será descartada na hora de armazenar em `a` novamente.

Para imprimir o valor da variável inteira, podemos utilizar a função `printf` da seguinte maneira:

```
1 #include <stdio.h>
2 int main() {
3     int a;
4     a = 2; // a vale 2
5     printf("a = %d\n", a);
6     a = a + 3; // a vale 5
7     printf("a = %d\n", a);
8 }
```

A saída desse programa será:

```
1 2
2 5
3
```

## Variável Float

A variável do tipo `float` é capaz de representar uma aproximação de um número real no computador.

```
1 #include <stdio.h>
2 int main() {
3     float a;
4     a = 0.25; // a vale 2
5     a = a + 3; // a vale 3.25
6     a = a * 2; // a vale 6.5
7 }
```

Para imprimir o valor da variável inteira, podemos utilizar a função printf da seguinte maneira:

```
1 #include <stdio.h>
2 int main() {
3     float a;
4     a = 0.25; // a vale 0.25
5     printf("a = %f\n", a);
6     a = a * 2; // a vale 0.5
7     printf("a = %f\n", a);
8 }
```

A saída desse programa será:

```
1 a = 0.25
2 a = 0.5
3
```

## Fahrenheit para Celsius

Faça um programa que lê uma temperatura em graus Fahrenheit e retorne a temperatura em graus Centígrados. Obs:

$$C = 5/9 \times (F - 32) \quad (1)$$

```
1 #include <stdio.h>
2
3 int main(){
4     float C, F;
5     printf("Entre com a temperatura em Fahrenheit: ");
6     scanf("%f", &F);
7     C = 5.0/9.0*(F - 32);
8     printf("Celsius = %f\n", C);
9     return 0;
10 }
```

## Celsius para Fahrenheit

Faça um programa que lê uma temperatura em graus Celsius e retorne a temperatura em graus Fahrenheit. Obs:

$$F = 9/5 \times C + 32 \quad (2)$$

```
1 #include <stdio.h>
2
3 int main(){
4     float C, F;
5     printf("Entre com a temperatura em Fahrenheit: ");
6     scanf("%f", &F);
7     F = (9.0*C/5.0) + 32;
8     printf("Celsius = %f\n", C);
9     return 0;
10 }
```

## Atenção

As expressões aritmética abaixo possuem valores diferentes:

```
1 F = 9.0*C/5.0 + 32;
2 F = 9.0/5.0*C + 32;
```

## Soma dígitos

Faça um programa que lê um número de até 3 dígitos (1 até 999) e devolve a soma dos seus dígitos.

```
1 #include <stdio.h>
2
3 int main(){
4     int a, c, d, u;
5     scanf("%d", &a);
6     c = a / 100;
7     a = a - 100*c;
8     d = a / 10;
9     a = a - 10*d;
10    u = a;
11    printf("soma de dígitos é ", c+d+u);
12 }
```

A linguagem C possui um operador que obtém o resto da divisão.

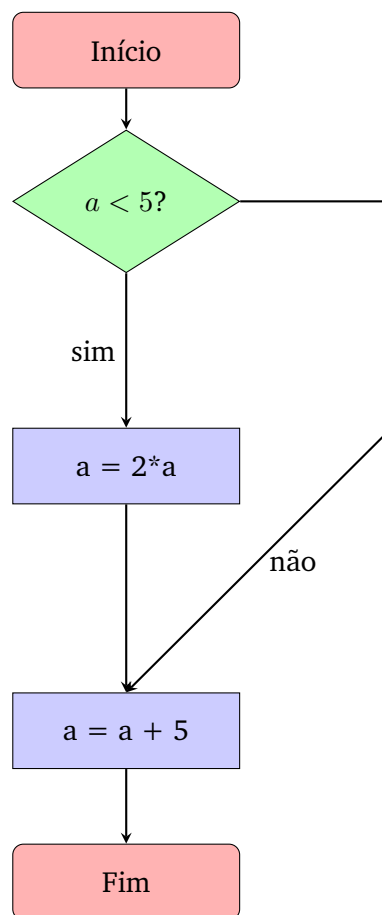
```
1 #include <stdio.h>
2
3 int main(){
4     int a, c, d, u;
5     scanf("%d", &a);
6     c = a / 100;
7     a = a%100;
8     d = a / 10;
9     a = a%10;
10    u = a;
11    printf("soma de dígitos é ", c+d+u);
12 }
```

## Desvio Condicional

Em um programa, podemos definir condições para controlar a execução de certos comandos. Suponha que queremos dobrar o valor de a apenas se ele for menor que 5 e, independentemente do resultado dessa verificação, adicionar 5 ao valor de a. Esse comportamento pode ser implementado com o seguinte código:

```
1 if(a<5){
2     a = 2*a;
3 }
4 a = a + 5;
```

Podemos representar graficamente, esse trecho de código da seguinte maneira:

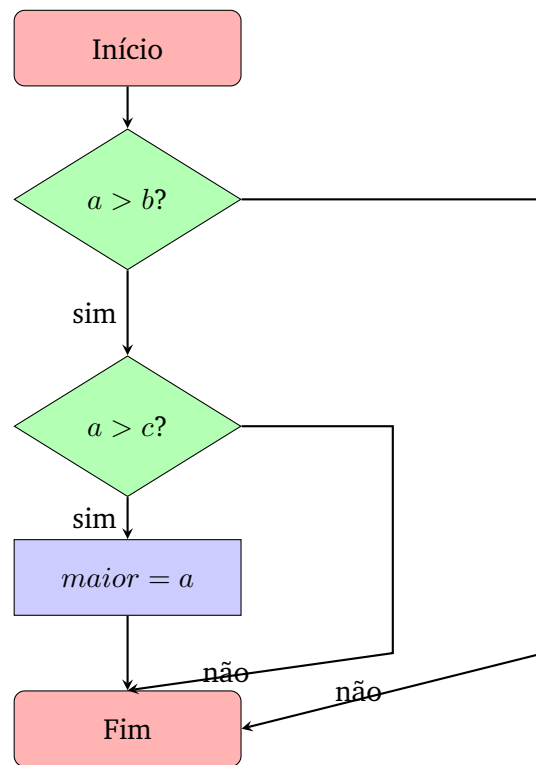


No fluxograma, os retângulos azuis representam instruções que serão executadas, enquanto o losango verde indica um ponto de decisão no fluxo do programa. Se a condição  $a < 5$  for verdadeira, o valor de  $a$  será dobrado antes de somarmos 5. Caso contrário, passamos diretamente para a soma de 5.

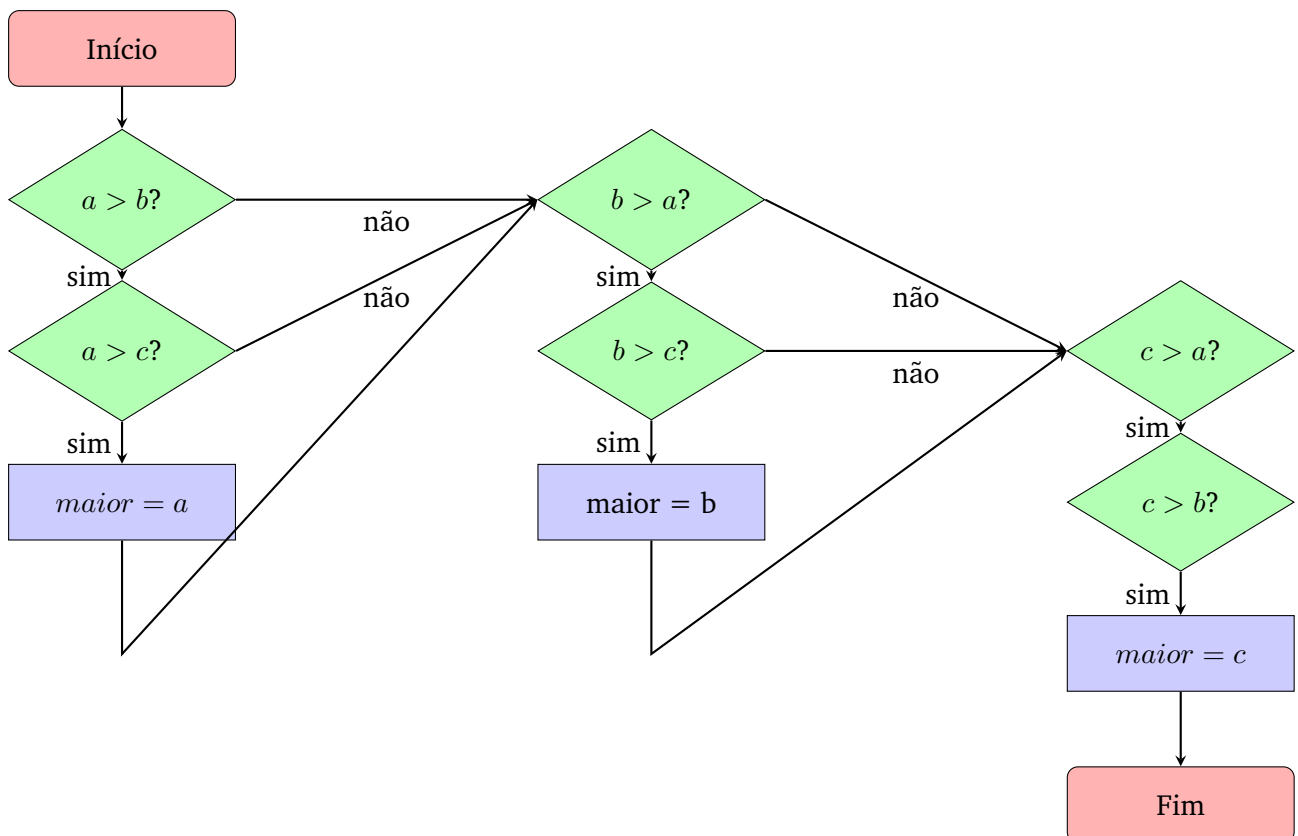
## Maior de 3 números

Imagine que nós temos três valores distintos  $a$ ,  $b$  e  $c$  e queremos ensinar o computador a identificar a maior entre eles utilizando o operador  $>$ .

Nosso programa, pode ter três resultados possíveis. A nossa primeira tentativa seria construir o caminho para identificar que  $a$  é o maior.



Note se  $a > b$  ou  $a > c$  forem falsas, o programa já vai se encaminha para a caixinha do fim. O caminho para identificar que o  $b$  é o maior e  $c$  é o maior é semelhante. Agora, podemos juntar os trechos da seguinte forma:



Na linguagem C, o programa acima ficaria da seguinte maneira:

```

1  if(a>b){
2      if(a>c){
3          maior = a;
4      }
5  }
6  if(b>a){

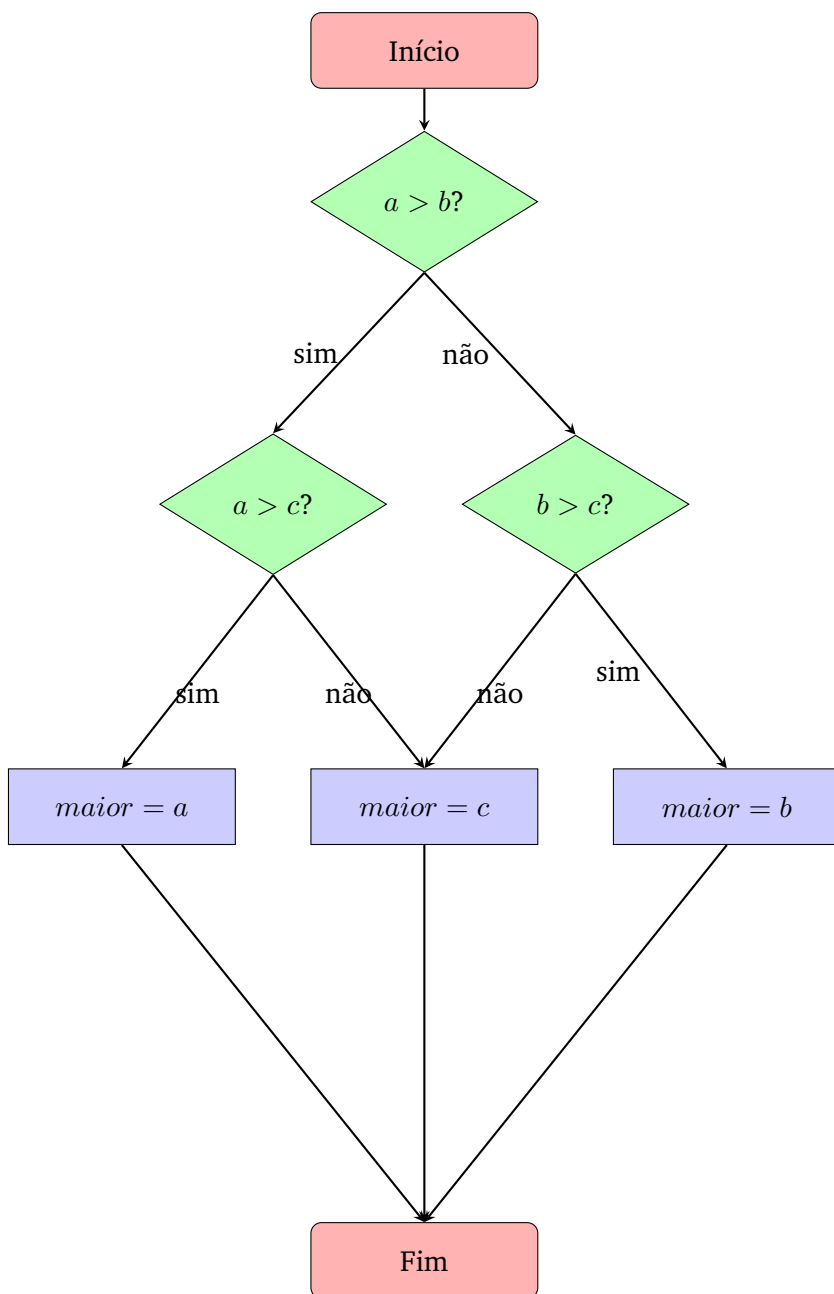
```

```

7   if(b>c){
8       maior = b;
9   }
10  }
11  if(c>a){
12      if(c>b){
13          maior = c;
14      }
15  }

```

Note que esse programa não está aproveitando os resultados dos testes. Por exemplo, se  $a > b$  é falso e eles são diferentes então  $b > a$ . Essa informação poderia ser aproveitada de alguma forma.



```

1  if(a>b){
2      if(a>c){
3          maior = a;
4      }else{
5          maior = c;
6      }
7  }else{
8      if(b>c){
9          maior = b;

```

```

10     }else{
11         maior = c;
12     }
13 }

```

## Usando conectivos lógicos

Podemos reescrever o primeiro programa de forma mais concisa utilizando operadores lógicos:

```

1  if(a>b && a>c) maior = a;
2  if(b>a && b>c) maior = b;
3  if(c>a && c>b) maior = c;

```

No entanto, uma abordagem mais eficiente evita verificações desnecessárias:

```

1  if(a>b && a>c) maior = a;
2  else if(b>c) maior = b;
3  else maior = c;

```

Observe que, se a condição  $a > b \ \&\& \ a > c$  for falsa, significa que  $a$  não é o maior valor. Dessa forma, basta verificar se  $b$  é maior que  $c$ ; caso contrário,  $c$  será o maior número.

## Exercícios

1. Faça um programa que lê dois pontos  $P_1 = (x_1, y_1)$  e  $P_2 = (x_2, y_2)$  e devolva a distância entre estes dois pontos, cujo valor é impresso no programa principal.

$$dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

2. Escreva um programa que receba três dígitos decimais  $d_2, d_1, d_0$ , representando as centenas, dezenas e unidades de um número de três dígitos. O programa deve reconstruir e exibir o número correspondente.
3. Escreva um programa que receba quatro dígitos binários  $b_3, b_2, b_1, b_0$  e exiba o número decimal correspondente.
4. Faça um programa que recebe um número de 1 até 15 e encontre e mostre seus dígitos binários.
5. Faça um programa que lê 4 números inteiros  $p, q, r$  e  $s$  representando as frações  $p/q$  e  $r/s$ . Informe a maior dessas frações.
6. Faça um programa que leia 3 números e imprima esses três números em ordem.

Pensando um pouquinho tem 6 casos:

- $a < b < c$
- $b < a < c$
- $c < a < b$
- $a < b < c$
- $b < c < a$
- $c < b < a$

7. (Professor Bonzinho) O professor decidiu aplicar 3 provas ao longo do semestre. Mas, ela só vai considerar as duas maiores notas no cálculo da média.

Se a média for maior igual a 7, o aluno é considerado aprovado. Caso contrário, o aluno está reprovado.

Faça um programa que implementa essa lógica, e imprime a situação final do estudante na tela: Aprovado ou Reprovado.



8. (Média Ponderada) As vezes, nós queremos calcular a média entre dois números  $x$  e  $y$ , onde um deles é mais importante do que o outro.

Para fazer isso, nós associamos um peso a cada número.

Por exemplo, se  $x$  é duas vezes mais importante do que  $y$ , a gente define os pesos assim  $\text{pesoX}=2$  e  $\text{pesoY}=1$

E daí, a gente calcula a média assim

$$\text{MediaPonderada} = \frac{x \times \text{pesoX} + y \times \text{pesoY}}{\text{pesoX} + \text{pesoY}} \quad (4)$$

Essa média é chamada de média ponderada.

Faça um programa que lê

- lê os valores de  $x$  e  $y$  do teclado
- lê os pesos associados a  $x$  e  $y$  do teclado
- calcula e imprime a média ponderada de  $x$  e  $y$  na tela

#### 9. Classificação de Notas

Uma escola utiliza um sistema de letras para classificar o desempenho dos alunos com base em suas notas numéricas. A conversão das notas segue a tabela abaixo: int

Intervalo de Notas	Classificação
$90 \leq \text{nota} \leq 100$	A
$80 \leq \text{nota} < 90$	B
$70 \leq \text{nota} < 80$	C
$60 \leq \text{nota} < 70$	D
$\text{nota} < 60$	F

Escreva um programa que recebe uma nota inteira  $N$  ( $0 \leq N \leq 100$ ) e retorna a classificação correspondente.

10. (Gangorra) Seja  $P1$  e  $P2$  são os pesos da criança no lado esquerdo e direito, respectivamente, e  $C1$  e  $C2$  são os comprimentos da gangorra do lado esquerdo e direito, respectivamente. A gangorra estará equilibrada quando

$$P1 * C1 = P2 * C2 \quad (5)$$

Seu programa deve produzir uma única linha, contendo um único inteiro. Se a gangorra estiver equilibrada, imprima "0". Se ela estiver desequilibrada de modo que a criança esquerda esteja na parte de baixo, imprima "-1", senão, imprima "1".

11. (Tabela) Considere a seguinte tabela:

linha/coluna	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
⋮	⋮	⋮	⋮	⋮

Faça um programa que dado um número inteiro  $N > 0$ , calcule a linha e a coluna em que esse número vai aparecer nessa tabela. Por exemplo, o número 10 aparece na linha 2 e coluna 2.