

1º Semestre

Técnicas de Programação

Luiz Fernando Carvalho

luizfcarvalhoo@gmail.com

Algoritmos Computacionais

Lógica de Programação



Linguagem de Programação

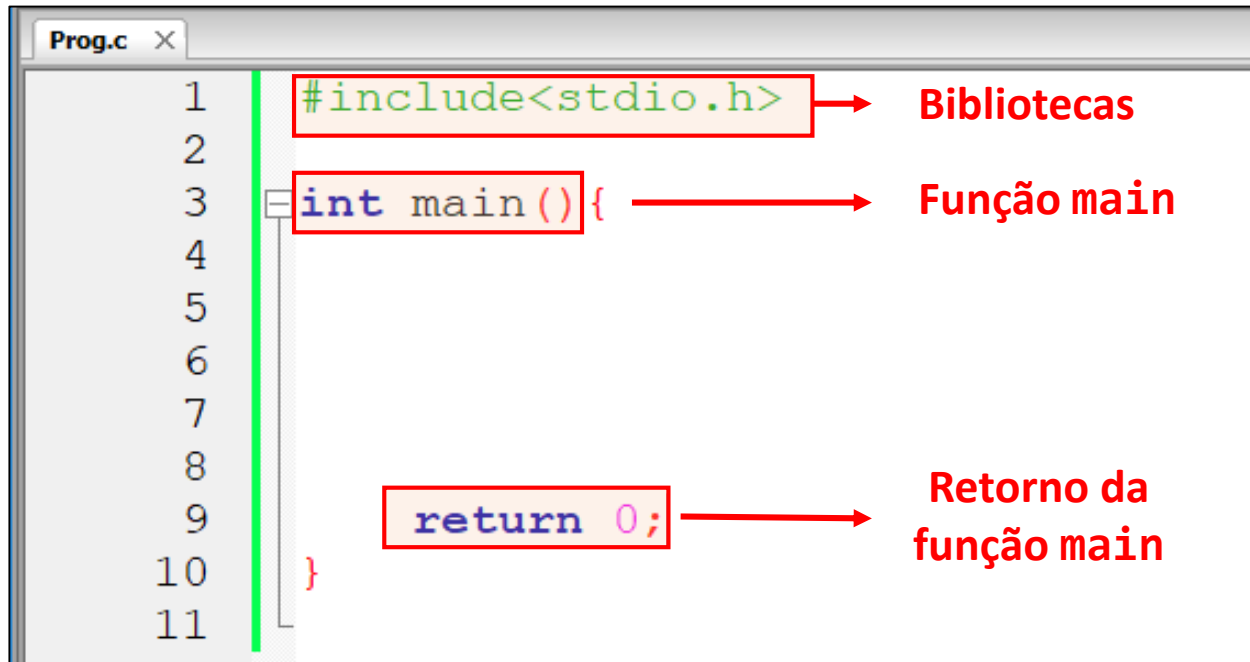


Aplicação/Programa



Programador
formula a solução

Estrutura básica em C



The image shows a code editor window titled 'Prog.c' with a line number margin on the left (1-11). The code is as follows:

```
1 #include<stdio.h>
2
3 int main(){
4
5
6
7
8
9
10 }
11
```

Annotations with red boxes and arrows:

- Line 1: `#include<stdio.h>` is boxed and labeled **Bibliotecas**.
- Line 3: `int main(){` is boxed and labeled **Função main**.
- Line 9: `return 0;` is boxed and labeled **Retorno da função main**.

Indica o fim da execução da função main

A execução do programa inicia-se pela função main.

Bibliotecas contém conjunto de funções e instruções previamente estabelecidas e que podem ser usadas pela programa

Declaração de Variáveis

- Uma variável sempre deve ser declarada antes da sua execução.
 - Declarar indica ao compilador para reservar um espaço na memória para guardar os valores dessa variável;

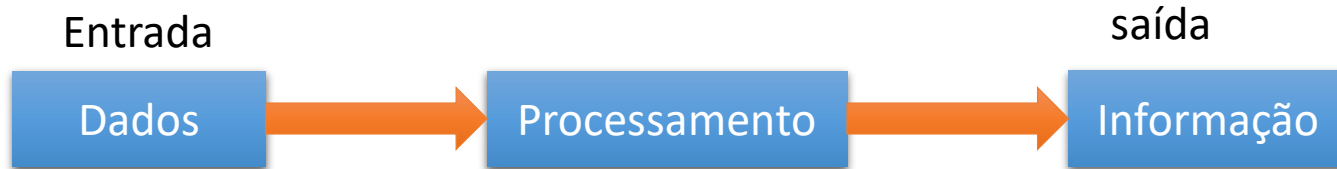
Exemplos de declaração

```
int main(){  
  
int num1;  
int num2;  
float resultado;  
float multiplicação;  
.  
.  
.
```

=

```
int main(){  
  
int num1, num2;  
float resultado, multiplicação;  
.  
.  
.
```

Linguagem C



Entrada (IN):

```
scanf(formato, &variável);
```

Exemplos:

```
scanf("%f", &n);  
scanf("%d", &num);  
scanf("%d", &cont);
```

Saída (OUT):

```
printf(conteúdo, [parâmetros]);
```

Exemplos:

```
printf("%f", n);  
printf("Exame");  
printf("O numero e %d", num);  
printf("sao necessárias %d moedas", cont);
```

Entrada e Saída de Dados

- Saída de dados (Imprimir)

- Sintaxe Básica

`printf("Mensagem escrita na tela");`

Formatação:

`\t` → espaço de tabulação

`\n` → quebra de linha

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5
6     printf("Hello World");
7
8     return 0;
9
10 }
```

"C:\Users\Fernando\Desktop\programa 2\teste.exe"

Hello World
Process returned 0 (0x0) execution time : 0.027 s
Press any key to continue.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5
6     printf("Hello World\n");
7
8     return 0;
9
10 }
```

"C:\Users\Fernando\Desktop\programa 2\teste.exe"

Hello World
Process returned 0 (0x0) execution time : 0.027 s
Press any key to continue.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5
6     printf("\tHello World");
7
8     return 0;
9
10 }
```

"C:\Users\Fernando\Desktop\programa 2\teste.exe"

Hello World
Process returned 0 (0x0) execution time : 0.028 s
Press any key to continue.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5
6     printf("\tHello World\n");
7
8     return 0;
9
10 }
```

"C:\Users\Fernando\Desktop\programa 2\teste.exe"

Hello World
Process returned 0 (0x0) execution time : 0.026 s
Press any key to continue.

Entrada e Saída de Dados

- Sintaxe geral

```
printf("Mensagem escrita na tela", lista_argumentos);
```

Exemplo:

```
int num = 5;  
char letra = 'C';  
float pi = 3.1415;
```

```
printf("O valor de num e' %d", num);
```

```
printf("O valor de num e' %d e o valor de letra e' %c", num, letra);
```

```
printf("num = %d, letra = %c e pi = %f", num, letra, pi);
```

%c → caractere (*char*)

%d → inteiro (*int*)

%f → ponto flutuante (*float*)

Entrada e Saída de Dados

Códigos usados na função `printf()`

Código	Formato
%c	Caractere
%d	Inteiro decimal com sinal
%e	Notação científica
%f	Ponto flutuante decimal
%g	Menor representação entre %f e %e
%o	Octal sem sinal
%s	<i>String</i> de caracteres
%u	Inteiros decimais sem sinal (<i>unsigned int</i>)
%x	Hexadecimal sem sinal
%%	Símbolo %

Entrada e Saída de Dados

- **Formato de impressão dos números reais**
- Junto com o **%f** uma especificação de quantas casas decimais se deseja que o número tenha;
- Especifica-se também o número total de caracteres do número a ser impresso.

```
printf(“%6.3f”, x);
```

especifica que se quer imprimir um **float** com **3 casas decimais** e com um **tamanho total** de **6** caracteres no total.

Entrada e Saída de Dados

- **Formato de impressão dos números reais**

```
printf(“%6.3f”, x);
```

especifica que se quer imprimir um **float** com **3 casas decimais** e com um **tamanho total** de **6** caracteres no total.

- o número de casas decimais é sempre respeitado;
- o **tamanho total** inclui o ponto decimal e um eventual sinal de menos (-);
- Se a soma do número de caracteres da **parte inteira**, mais o **ponto decimal**, mais a **parte fracionária**, mais um **eventual sinal de menos** *ainda for menor* do que o tamanho total especificado no formato, então, espaços em branco serão acrescentados à esquerda da parte real do número.
- Se o tamanho total de caracteres for maior do que o tamanho total especificado no formato, então, apenas o número de casas decimais é respeitado

Entrada e Saída de Dados

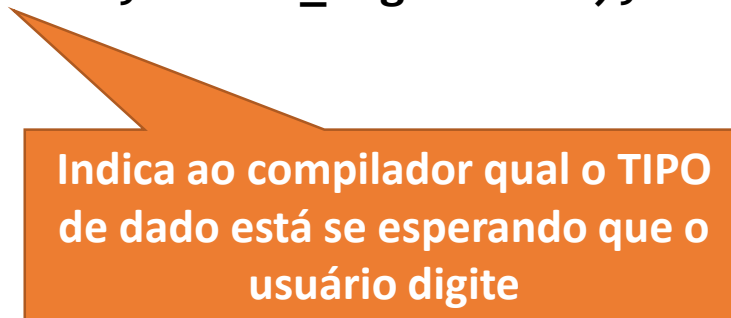
- Entrada de Dados
- Sintaxe

```
scanf(“formato”, lista_argumentos);
```

Exemplo:

```
int num;  
char letra;  
float pi;
```

```
scanf(“%d”, &num);  
scanf(“%c”, &letra);  
scanf(“%f”, &pi);  
scanf(“%d %c %f”, &num, &letra, &pi);
```



Indica ao compilador qual o TIPO de dado está se esperando que o usuário digite

Linguagem C - Exemplo

```
#include<stdio.h>

int main(){
    int idade;

    printf("digite sua idade: ");
    scanf("%d", &idade);

    printf("Daqui 5 anos você terá %d anos", idade + 5);

    return 0;
}
```

Exercícios

- Crie um programa para encontrar as raízes de uma equação do segundo grau. Considere que os coeficientes podem ser decimais. Não é necessário tratar $\Delta < 0$.
 - Teste para $x^2 - 2x - 3 = 0 \rightarrow x' = 3; x'' = -1$
- Escreva um programa que lê uma temperatura em graus Fahrenheit, calcular e escrever o valor correspondente em graus Celsius (baseado na fórmula abaixo):

$$\frac{C}{5} = \frac{F - 32}{9}$$