

Introdução à Linguagem C e Práticas de Programação

- O código está funcionando?
- Sim perfeitamente.

O código:



Compiladores/IDEs (sugestões):

Linguagem de programação adotada na disciplina: C

- <https://replit.com/languages/c>
- <https://www.onlinegdb.com>
- Dev-C++
- Code::Blocks
- Visual C++ (Windows)
- GCC (Linux/MAC)

1) <https://replit.com/languages/c>

C online editor, IDE, compiler, interpreter, and REPL

Code, collaborate, compile, run, share, and deploy C and more online from your browser

Sign up for the full experience

C

Run



Share

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

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
>
```

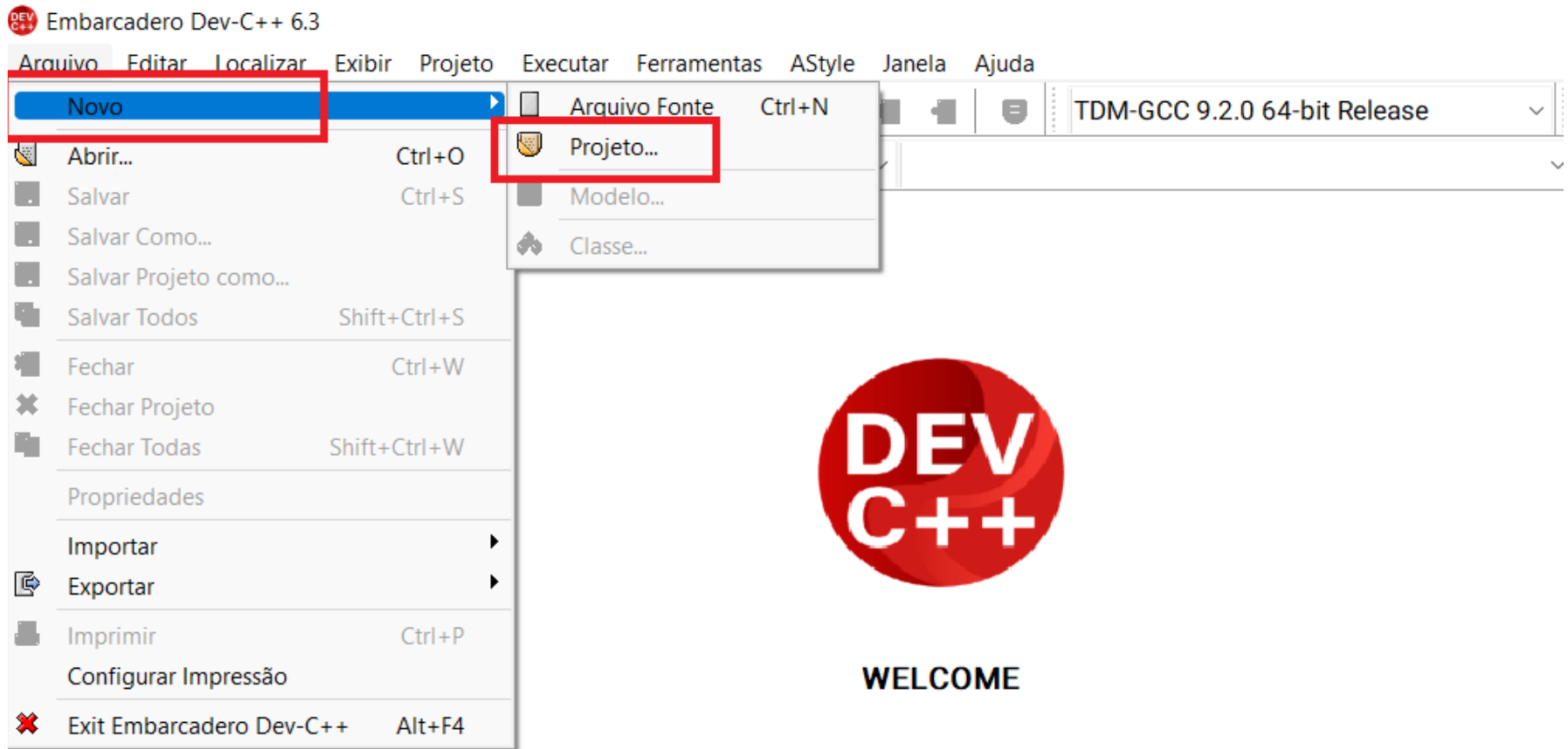
2) Dev-C++: Tutoriais de instalação

1. <http://linguagemc.com.br/tutorial-de-instalacao-do-dev-c/>
2. https://www.inf.pucrs.br/flash/cbp/instala_dev/
3. <https://www.inf.pucrs.br/~pinho/Laprol/DevC/Dev.html>
(como manusear projetos/código-fontes no Dev)
4. Tutoriais no Youtube

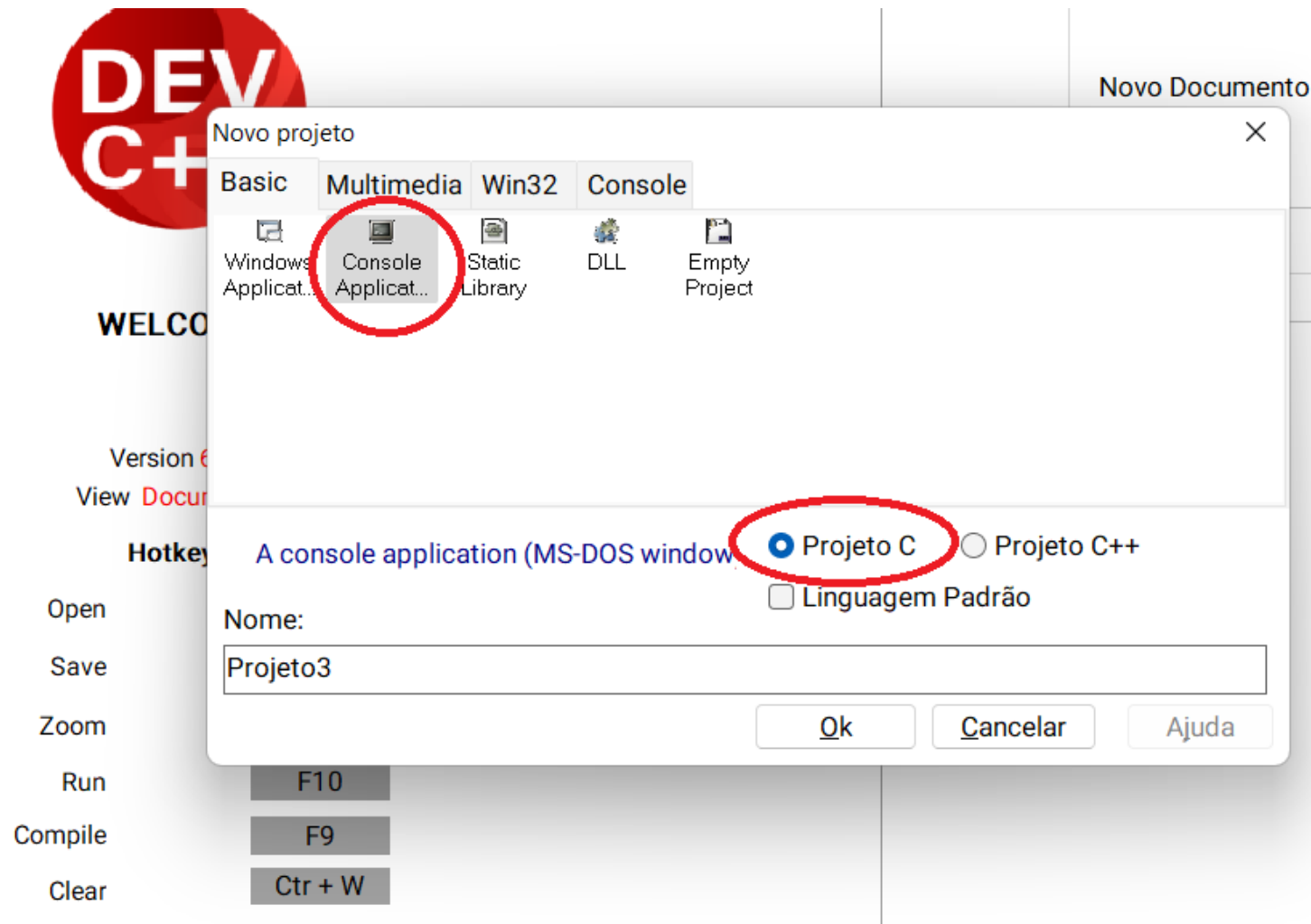
Versões do Dev-C++:

- Instalada no laboratório: Versão 5.11
- Última versão disponível no site: Versão 6.3

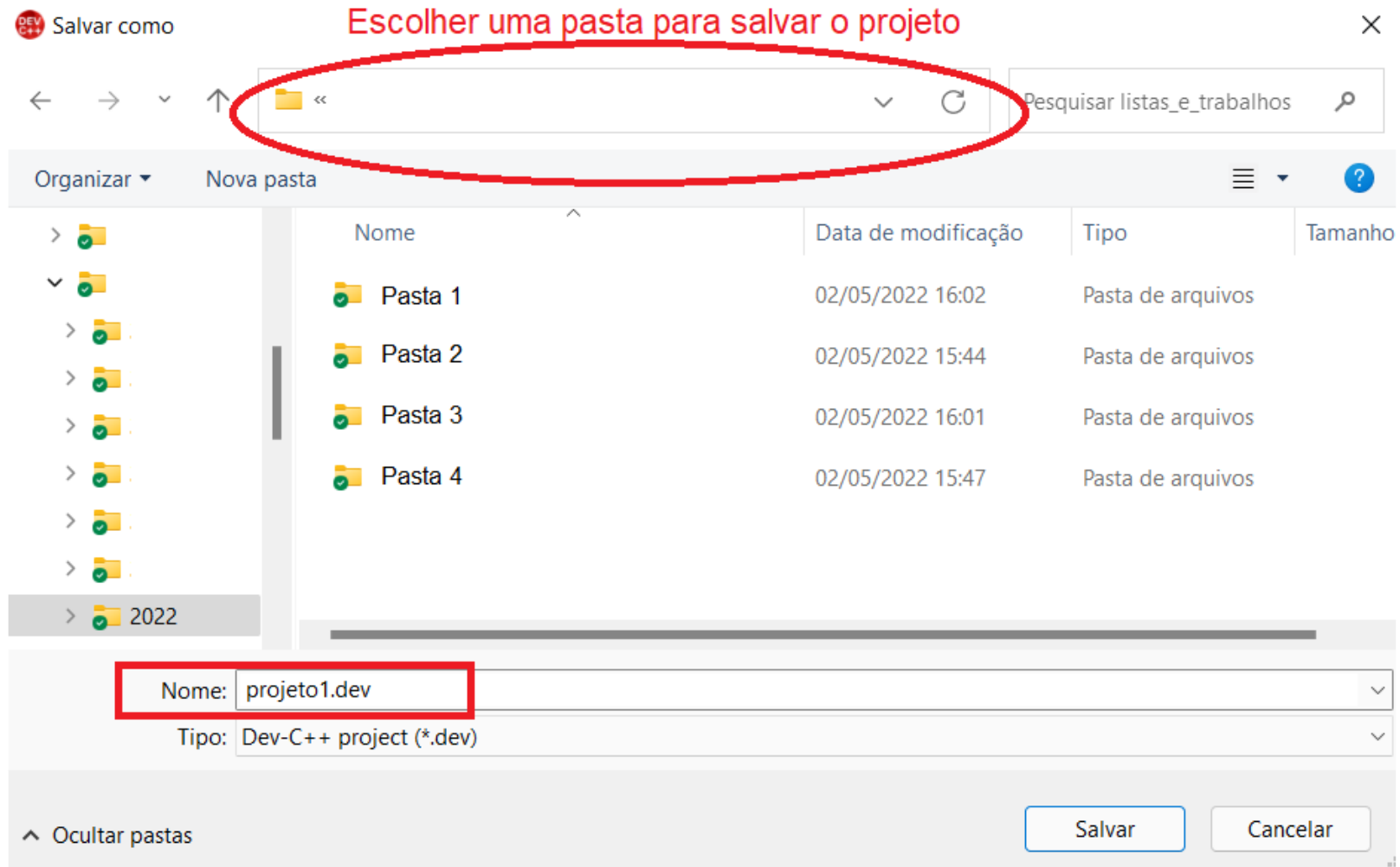
Dev-C++: criando um novo projeto



Dev-C++: criando um novo projeto



Dev-C++: criando um novo projeto



3) Visual Studio: Tutoriais de instalação

1. <https://visualstudio.microsoft.com/pt-br/vs/getting-started/>
2. https://www.youtube.com/watch?v=iq_sBeiN-Qk

3. Tutoriais no Youtube

Versões do Visual Studio Code:

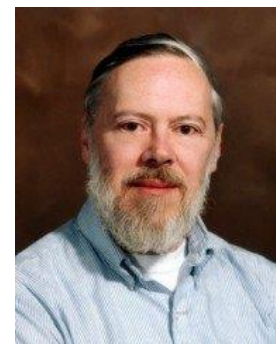
- Instalada no laboratório: 1.66 e 2022 Community

Perguntas?



VISÃO GERAL - C

- Inventada e implementada por Dennis Ritchie em um sistema UNIX (sistema base do Linux) no início da década de 70.
- ANSI (American National Standards Institute) estabeleceu, em 1983, um padrão para a linguagem C.

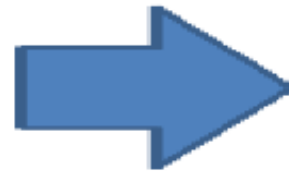


COMPILADOR

- Um compilador faz a conversão do código fonte para o *código-objeto*.
- **Código objeto:** é uma “tradução” do código fonte do programa, de forma que o computador possa executar o programa diretamente a partir de um código binário (de máquina).
- A linguagem C é estruturada e compilada.

DIAGRAMATIZAÇÃO

```
#include <stdio.h>
int main ( )
{
    printf("Oi mundo!!\n");
    return 0;
}
```



Compilador



```
00000111111111
1101010101010
1010101010010
0101001010110
```



PASSO-A-PASSO PARA PROGRAMAR EM C

1. Escrever o algoritmo (programa) usando os comandos da linguagem C.
2. Compilar o programa.
3. Ligar o programa com as funções necessárias da biblioteca (*linkagem*).
4. Gera-se, então, o programa final de computador.

COMPILANDO NO C

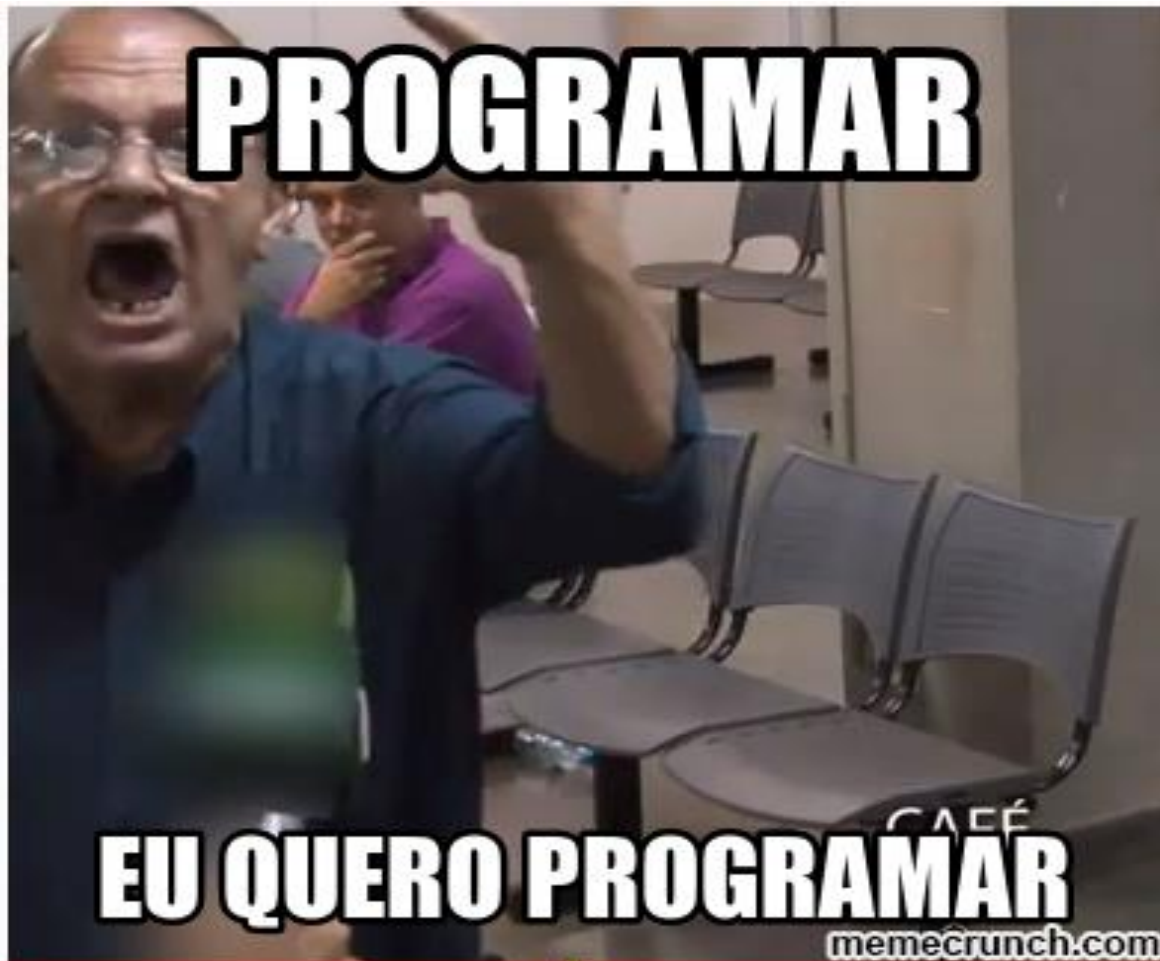
- Todo compilador C vem com um conjunto padrão de **bibliotecas de comandos** (funções) que realizam as tarefas necessárias mais comuns.
- Várias funções para utilização já estão nessa biblioteca padrão.
- O compilador que usaremos é o **GCC** (**GNU Compiler Collection**).

Meu primeiro programa

- Meu primeiro programa: famoso “Hello World !!!”



APRENDENDO C COM EXEMPLOS



COMANDO `main()`

- ❑ O comando `main()` define uma função de nome `main`.
- ❑ Todos os programas em C têm que ter uma função `main`, pois são os comandos que estão nela que serão executados quando o programa for rodado.

❑ Analogia:

```
int main()
```

```
{ ← abre chave = “início” do algoritmo
```


```
} ← fecha chave = “fim” do algoritmo
```

EXEMPLO

```
//Algoritmo Soma de dois Números
#include <stdio.h>
int main()
{
    //Variáveis
    float x, y, soma;

    //Programa principal
    printf("Primeiro numero: ");
    scanf("%f", &x);
    printf("Segundo numero: ");
    scanf("%f", &y);

    soma = x + y;
    printf("Soma dos dois numeros: %.2f\n", soma);
    return 0;
}
```



O conteúdo da função é delimitado por chaves

RETORNO DA FUNÇÃO `MAIN()`

- Convenção:

- ▣ Se o programa retornar zero, significa que ele terminou normalmente.

- ▣ Se o programa retornar um valor diferente de zero, significa que ele teve um termino anormal.

- Futuramente vamos falar mais sobre funções.

EXEMPLO

```
//Algoritmo Soma de dois Números
#include <stdio.h>
int main()
{
    //Variáveis
    float x, y, soma;

    //Programa principal
    printf("Primeiro numero: ");
    scanf("%f", &x);
    printf("Segundo numero: ");
    scanf("%f", &y);

    soma = x + y;

    printf("Soma dos dois numeros: %.2f\n", soma);
    return 0;
}
```

EXEMPLOS

```
//Algoritmo Soma de dois Números
```

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    //Variáveis
```

```
    float x, y, soma;
```

```
    //Programa principal
```

```
    printf("Primeiro numero: ");
```

```
    scanf("%f", &x);
```

```
    printf("Segundo numero: ");
```

```
    scanf("%f", &y);
```

```
    soma = x + y;
```

```
    printf("Soma dos dois numeros: %.2f\n", soma);
```

```
}
```

**Função main (sem
retorno)!**



ESTRUTURA BÁSICA DE UM PROGRAMA

EXEMPLOS

```
//Algoritmo Soma de dois Números
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
//Variáveis
```

```
float x, y, soma;
```

```
//Programa principal
```

```
printf("Primeiro numero: ");
```

```
scanf("%f", &x);
```

```
printf("Segundo numero: ");
```

```
scanf("%f", &y);
```

```
soma = x + y;
```

```
printf("Soma dos dois numeros: %.2f\n", soma);
```

```
return 0;
```

```
}
```

**Comentários são
delimitados por //
ou /* blabla */**

ESTRUTURA BÁSICA DE UM PROGRAMA

EXEMPLOS

```
//Algoritmo Soma de dois Números
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
//Variáveis
```

```
float x, y, soma;
```

```
//Programa principal
```

```
printf("Primeiro numero: ");
```

```
scanf("%f", &x);
```

```
printf("Segundo numero: ");
```

```
scanf("%f", &y);
```

```
soma = x + y;
```

```
printf("Soma dos dois numeros: %.2f\n", soma);
```

```
return 0;
```

```
}
```

Comentários são delimitados por //
ou /* blabla */

Observe a indentação (recuo)!!!



TIPOS DE DADOS EM C

Tipo	Bits	Escala
<code>int</code> (inteiro)	32	-2.147.483.648 a 2.147.483.647
<code>short</code> (ou <code>short int</code>)	16	-32.768 a 32.767
<code>long</code> (ou <code>long int</code>)	32	-2.147.483.648 a 2.147.483.647
<code>float</code> (real)	32	$3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
<code>double</code> (real)	64	$1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
<code>long double</code>	80	$3,4 \times 10^{-4932}$ a $3,4 \times 10^{4932}$
<code>char</code> (caracter)	8	-128 a 127
<code>unsigned char</code>	8	0 a 255
<code>unsigned int</code>	32	0 a 4.294.967.295
<code>unsigned short</code>	16	0 a 65.535
<code>unsigned long</code>	32	0 a 4.294.967.295

DECLARAÇÃO DE VARIÁVEIS

```
int a;
```

```
float num;
```

```
double n1, n2, n3;
```


Inicialização

```
int a = 10;  
char c = 'm' ;
```

OPERADORES ARITMÉTICOS EM C

+	Adição ou positivo
-	Menos ou negativo
*	Multiplicação
/	Divisão
%	Resto da divisão inteira ou mod
=	Atribuição

OPERADORES ARITMÉTICOS

EXEMPLOS

```
//Algoritmo Soma de dois Números
#include <stdio.h>
main()
{
    //Variáveis
    float x, y, soma;

    //Programa principal
    printf("Primeiro numero: ");
    scanf("%f", &x);
    printf("Segundo numero: ");
    scanf("%f", &y);

    soma = x + y;

    printf("Soma dos dois numeros: %.2f\n", soma);
}
```

 Operadores
Aritméticos

ARQUIVOS-CABEÇALHOS (HEADERS)

- Para usar alguns comandos, precisamos explicitamente incluir suas definições por meio do `#include <...>`
- Nesta aula, para usarmos os comandos `printf` e `scanf`, precisamos colocar no início a declaração:
- `#include <stdio.h>`
- Conforme precisarmos de **novos comandos**, vamos aprendendo cada um dos **includes**.

COMANDO PRINTF

□ O comando de saída (escreva) de pseudocódigo é, na verdade, o `printf` em C.

■ Saída de dados:

–no algoritmo: `printf("Eu gosto de C!\n");`

–na tela: Eu gosto de C!

COMANDO PRINTF

□ `printf("Soma de dois números: %.2f\n", soma);`

■ Caracteres Especiais:

- `\n`: nova linha (enter)
- `\t`: tab
- `\\`: barra \
- `\"`: aspas duplas
- `\'`: aspas simples
- `\(`: abre parênteses
- `\)`: fecha parênteses

COMANDO PRINTF

□ `printf("Soma de dois números: %.2f\n", soma);`

■ Exibição de Variáveis:

- `%d` : int
- `%f` : float
- `%c` : char
- `%lf` : double
- `%ld` : long (ou long int)
- `%.2f` : 2 dígitos decimais

COMANDO SCANF

- O comando de entrada (leia) de pseudocódigo é, na verdade, o `scanf` em C.
- Entrada de dados:
 - no algoritmo: `scanf ("%f", &x) ;`
 - na tela: cursor piscando esperando que algum dado seja digitado a partir do teclado

EXERCÍCIO 1

- Dados os valores do raio de uma circunferência, e tomando **$\pi = 3.141592$** , faça um programa em C que calcula o valor da área do círculo.

```
#include <stdio.h> //Biblioteca para uso das funções de entrada/saída
```

```
// Programa: Exercício 1
```

```
int main()
```

```
{
```

```
    //Declaracao de variaveis
```

```
    //Rotina principal
```

```
}
```

EXERCÍCIO 2

- Faça um algoritmo que leia uma temperatura (em Fahrenheit), e a apresente convertida em graus Celsius.

- A fórmula de conversão é

$$C = (F - 32) * (5 / 9),$$

em que F é a temperatura em Fahrenheit, e C é a temperatura em Celsius.

EXERCÍCIO 3

- Dado um número de três algarismos $N = CDU$ (em que C é o algarismo das centenas, D é o algarismo das dezenas e U o algarismo das unidades). Considere o número M , constituído pelos algarismos de N em ordem inversa, isto é, $M = UDC$.
- Gerar M a partir de N (ex: $N = 123 \rightarrow M = 321$).
- **Dica:** Usar operador que calcula o resto de uma divisão inteira: %