

# Algoritmos e Programação

## Aula 1

### Introdução

### Linguagem C

#### Linguagem C

- Nasceu na década de 70
- Inventor: Dennis Ritchie
- É uma linguagem genérica utilizada para a criação de programas diversos como processadores de texto, planilhas eletrônicas, sistemas operacionais, programas de comunicação, programas para a automação industrial, gerenciadores de bancos de dados, programas de projeto assistido por computador, programas para a solução de problemas da Engenharia, Física, Química e outras Ciências, etc ...

## Linguagem C

- Particularidades:
  - Case sensitive
    - Maiúsculas e minúsculas fazem diferença: soma será diferente de Soma, SOMA, SoMa ou sOmA.
  - Comandos também são case sensitive: if e for

## Linguagem C

- Estrutura:
  - Um cabeçalho contendo as diretivas de compilador onde se definem o valor de constantes simbólicas, declaração de variáveis, inclusão de bibliotecas, declaração de rotinas, etc.
  - Um bloco de instruções principal e outros blocos de rotinas.
  - Documentação do programa: comentários.

## Linguagem C

- As linhas *#include* dizem ao compilador que ele deve incluir os arquivos-cabeçalho indicados.
- Nestes arquivos existem declarações de funções úteis para a execução do nosso programa.
- O C possui diversos arquivos-cabeçalhos. Alguns deles:

## Linguagem C

`#include <stdio.h>`

- leitura de dados digitados no teclado e exibição de informações na tela do programa

`#include <string.h>`

- funções para manipular strings

`#include <conio.h>`

- funções para manipular caracteres na tela, especificar cor de caracter e de fundo, etc.

`#include <dos.h>`

## Linguagem C

`#include <stdlib.h>`

- Funções envolvendo alocação de memória, controle de processos, conversões, etc.

`#include <windows.h>`

- Todas as definições de janelas: criar, abrir, etc.

`#include <time.h>`

- Contém protótipos e tipos de funções para manipular horários e datas

## Linguagem C

`#include <ctype.h>`

- Funções para conversão de maiúsculas, minúsculas e outros tratamentos de caracteres.

`#include <locale.h>`

- adaptação do programa às características de uma determinado idioma ou de uma região

`#include <math.h>`

- Fornece um conjunto de funções para operações matemáticas, tais como funções trigonométricas, hiperbólicas, logaritmos, potência e arredondamentos

## Linguagem C

### Utilizaremos:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <ctype.h>
#include <locale.h>
#include <math.h>
```

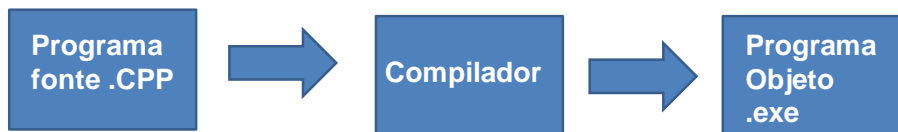
## Linguagem C

- Comentários
  - // Comentários em linha
  - /\* \*/ Todo intervalo entre os dois sinais
- Função *main()*
  - Função que é chamada quando o programa é executado
  - Delimitado por { }
  - O código que estiver dentro das chaves será executado sequencialmente quando a função for chamada.

## Linguagem C

- Compilação:
  - O código deve ser salvo em arquivo de texto com a extensão “.c” ou no caso do Dev-C, “.cpp”. Esse arquivo é o FONTE que é traduzido para a linguagem de máquina. Essa é a compilação, feita pelo programa chamado compilador. O Compilador lê o código-fonte e cria um arquivo executável (.exe).
  - No windows - .exe
  - No Linux – atributo “executável”

## Linguagem C



## Linguagem C

- Comando: printf (print formatted)
- Usado para imprimirmos o resultado do processamento ou para nos comunicar com o usuário do nosso programa.
- Forma geral:
  - `printf (string_de_controle, lista_de_argumentos);`

*String\_de\_controle*: Na string de controle indicamos quais, de qual tipo e em que posição estão as variáveis a serem apresentadas. Notação: %

## Linguagem C

- Para cada código de controle, teremos um argumento na lista de argumentos.
- Alguns dos códigos utilizados:

| Código    | Formato   |
|-----------|---|
| <b>%c</b> | <b>Um caracter (char)</b>                         |
| <b>%d</b> | <b>Um número inteiro decimal (int)</b>            |
| <b>%i</b> | <b>O mesmo que %d</b>                             |
| <b>%e</b> | Número com notação científica com o "e" minúsculo |
| <b>%E</b> | Número com notação científica com o "e" maiúsculo |
| <b>%f</b> | <b>Ponto flutuante decimal</b>                    |
| <b>%g</b> | Escolhe automaticamente o melhor entre %f e %e    |
| <b>%G</b> | Escolhe automaticamente o melhor entre %f e %E    |
| <b>%o</b> | Numero octal                                      |
| <b>%s</b> | <b>String</b>                                     |
| <b>%u</b> | Decimal "unsigned" (sem sinal)                    |
| <b>%x</b> | Hexadecimal com letras minúsculas                 |
| <b>%X</b> | Hexadecimal com letras maiúsculas                 |
| <b>%%</b> | <b>Imprime um %</b>                               |
| <b>%p</b> | Ponteiro  |

## Linguagem C

### Exemplos:

```
printf ("A cotacao do dolar hoje eh de %2.3f reais",
dolar);
```

```
printf ("Hoje estou comemorando %d anos", anos);
```

```
printf("Este eh o numero dois: %d", 2);
```

```
printf("\n\n%s esta a %d milhoes de milhas do sol",
"Venus", 67);
```



## Linguagem C

### Tamanho, justificação e número de casas decimais

- Códigos colocados entre o % e a letra que indica o tipo de formato.
- Um inteiro indica o tamanho mínimo, em caracteres, que deve ser reservado para a saída.
- Ex.: %5d – indica que o campo terá 5 caracteres de comprimento no mínimo. Se precisar de mais, o campo terá o tamanho necessário. Se for menos, terá 5 de comprimento e será preenchido com espaços em branco.
- Ex.: %05d: 5 casas para o número, se for menos, fará o preenchimento com zeros;
- Ex.: %-5d: Alinha à esquerda

## Linguagem C

### Tamanho, justificação e número de casas decimais

- Ponto Flutuante: %10.4f – comprimento total 10 com 4 casas decimais
- Quando utilizamos essa notação para inteiros strings, indica o número mínimo e máximo de casas.

Ex.: %5.8d - um inteiro com comprimento mínimo de cinco e máximo de oito.

Exemplos:

| CÓDIGO                   | IMPRIME |
|--------------------------|---------|
| Printf("%-5.2f",456.671) | 456.67  |
| printf ("%5.2f",2.671);  | 2.67    |
| printf ("%10s","Ola");   | Ola     |

## Linguagem C

- Regras para impressão de um número real
  - O número de casas decimais é sempre respeitado. Se for preciso, zeros serão acrescentados à direita do número;
  - O tamanho total significa o número de caracteres do número incluindo o ponto decimal e um eventual sinal de menos (-), se for o caso;
  - 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 a soma do número de caracteres da parte inteira, mais o ponto decimal, mais a parte fracionária, mais um eventual sinal de menos for maior do que o tamanho total especificado no formato, então, apenas o número de casas decimais é respeitado.

## Linguagem C

- Para deslocar em linhas na tela, utilizamos o comando `\n`
  - Ex.: `printf("\n\nMeu primeiro programa em C\n");`
- Podemos também utilizar o comando `printf` sem a string de controle e a lista de argumentos.
  - Por exemplo, para imprimirmos somente um texto podemos utilizar o comando conforme exemplos abaixo:
  - `printf("Estou aprendendo a programar");`