

# Entrada e saída de dados

[emanoelim@utfpr.edu.br](mailto:emanoelim@utfpr.edu.br)

# Entrada de dados

- Entrada de dados: permite pegar dados digitados pelo teclado e armazená-los em um variáveis.
- A principal forma de fazer a entrada de dados é pela função *scanf*.
- A forma padrão da leitura de dados é:

```
scanf("string de controle", argumentos);
```

# Entrada de dados

- Exemplos:

```
scanf("%d", &idade);
```

- “%d” é a string de controle. Ela indica o tipo de dado que vai ser guardado (nesse caso, int).
- **idade** é o nome da variável onde a informação digitada será guardada.
- A variável sempre deve estar precedida por um **&**, que quer dizer: “guarde o valor digitado no endereço de memória da variável x”.



# Entrada de dados

- Strings de controle:
  - %d ou %i - para tipo de dado int
  - %f - para tipo de dado float
  - %lf - para tipo de dado double
  - %c - para tipo de dado char
  - %s - cadeia de caracteres (palavras, textos - visto adiante).

# Saída de dados

- Saída de dados: permite mostrar informações na tela do computador.
- A principal forma de fazer a entrada de dados é pela função *printf*.
- A forma padrão da leitura de dados é:

```
printf ("texto, strings de controle e  
caracteres de formatação", lista de  
argumentos);
```

# Saída de dados

- **Texto:** a mensagem a ser mostrada;
- **Strings de controle:** se no meio do texto for necessário mostrar o valor de uma variável, no lugar onde deve ficar este valor é usada uma string de controle;
- **Caracteres de formatação:** indicam quebra de linha, tab, etc.
- **Argumentos:** variáveis cujo valor queremos mostrar.



# Saída de dados

- Exemplo de saída de dados somente com texto:

```
printf("Bom dia!");
```

- Exemplo de saída de dados mostrando o valor de uma variável:

```
int x = 10;  
printf("O valor de x é %d.", x);
```

No lugar da variável foi usada a string de controle “%d”, para indicar que ali seria mostrado um valor inteiro, o valor da variável “x”.

# Saída de dados

- É possível exibir o valor de várias variáveis no mesmo printf:

```
int x = 10;  
float y = 2.5;  
printf("O valor de x é %d e o valor de %f", x, y);
```

- No lugar de %d será mostrado o valor de x, no lugar de %f será mostrado o valor de y.
- As strings de controle e as variáveis devem estar na ordem correspondente.



# Saída de dados

- Exemplo com caracteres de formatação:

```
printf("Primeira frase \n");  
printf("Segunda frase");
```

- “\n” indica uma quebra de linha, ou seja, cada uma das frases será apresentada em uma linha diferente. Sem o “\n”, as duas frases ficariam “grudadas”.

# Saída de dados

- Caracteres de formatação disponíveis:
  - \n - nova linha
  - \t - tabulação (tab)
  - \b - retrocesso
  - \" - imprimir aspas
  - \\ - para imprimir barra
  - %% - para imprimir o símbolo de %

# Float e double - nº de casas

- O código abaixo terá como saída o número 2.500000:

```
...  
float x = 2.50;  
printf("x: %f \n", x);  
...
```

- Para delimitar o número de casas decimais, após o “%” da string de controle, devemos colocar “.” e o número de casas que queremos mostrar:

```
...  
float x = 2.50;  
printf("x: %.2f \n", x);  
...
```

Isso fará com que seja mostrado 2.50 em vez de 2.500000.



# Biblioteca stdio.h

- As funções *scanf* e *printf* permitem ler e imprimir dados de forma fácil e rápida. Nós não precisamos escrever um algoritmo para fazer isso.
- Estas funções já foram implementadas em uma biblioteca chamada “stdio.h”.
- Seu nome vem da expressão em inglês *standard input-output header*, que significa "cabeçalho padrão de entrada/saída".

# Biblioteca stdio.h

- Uma biblioteca é um arquivo que contém um conjunto de funções que podem ser importadas para o nosso código.
- Para que as funções *scanf* e *printf* funcionem corretamente, é preciso incluir este arquivo no nosso programa.
- A inclusão é feita através da diretiva “include”, ela instrui o compilador a ler outro arquivo e considerar o conteúdo deste arquivo.

# Biblioteca stdio.h

- Exemplo:

```
#include <stdio.h>

main() {
    int x;
    printf("Informe o valor para x:\n");
    scanf("%d", &x);
}
```

- O nome da biblioteca fica entre <>;
- Não vai ponto e vírgula;
- Deve ficar fora da main, no início do arquivo.



# Biblioteca stdio.h

- A linguagem C conta com várias bibliotecas padrão, que permitem utilizar diferentes tipos de funções.
- Outras bibliotecas serão vistas ao longo do semestre, conforme a necessidade.