



# TIPOS DE DADOS OPERADORES ENTRADA E SAÍDA DE DADOS

---

Bibliografia:

Cap 3 - Estudo Dirigido de Linguagem C

José Augusto Manzano, Ed. Erica

# Aula 1

## Objetivos

- Apresentação do prof. e do conteúdo programático
- Entender o arquivo contendo recomendações do professor
- Fazer o primeiro programa em C

# Estrutura de um programa em C

```
/* comentários, recomendo sempre colocar no topo o nome do programa, e o seu objetivo */
```

```
[<definições de pré-processamento – cabeçalhos>]
```

```
[<declaração das variáveis globais>]
```

```
int main ( [<parâmetros>])
```

```
{
```

```
    /* corpo da função, com declarações de variáveis locais,  
    comandos e funções de trabalho */
```

```
    return(0);
```

```
}
```

```
[
```

```
    [<tipo>] funçãonome ([<parâmetros>])
```

```
    [<declaração de parâmetros>]
```

```
{
```

```
    /* corpo da função, com declarações de variáveis locais,  
    comandos e funções de trabalho */
```

```
    [return ou return () ou return (valor)]
```

```
}
```

```
]
```

# Primeiro programa

```
/* 1.1.BemVindo
```

```
    Primeiro programa, que dá uma saldação em tela
```

```
*/
```

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

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    printf("Seja bem vindo a disciplina PC1 \n");
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

# Atividade

- Use a internet, e pesquise:
  - a) O que são o `stdio.h` e o `stdlib.h`?
  - b) O que faz o comando `printf`?
  - c) O que faz o `\n` no comando `printf`?
  - d) O que faz o `system("pause")`?
  - e) O que são linguagens compiladas e interpretadas
  - f) Quais as vantagens e desvantagens da linguagem C

# Fluxograma Símbolos

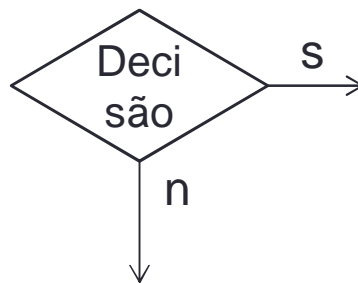
Início / Termina



conectores

Comandos

Entrada ou  
Saída de  
Dados

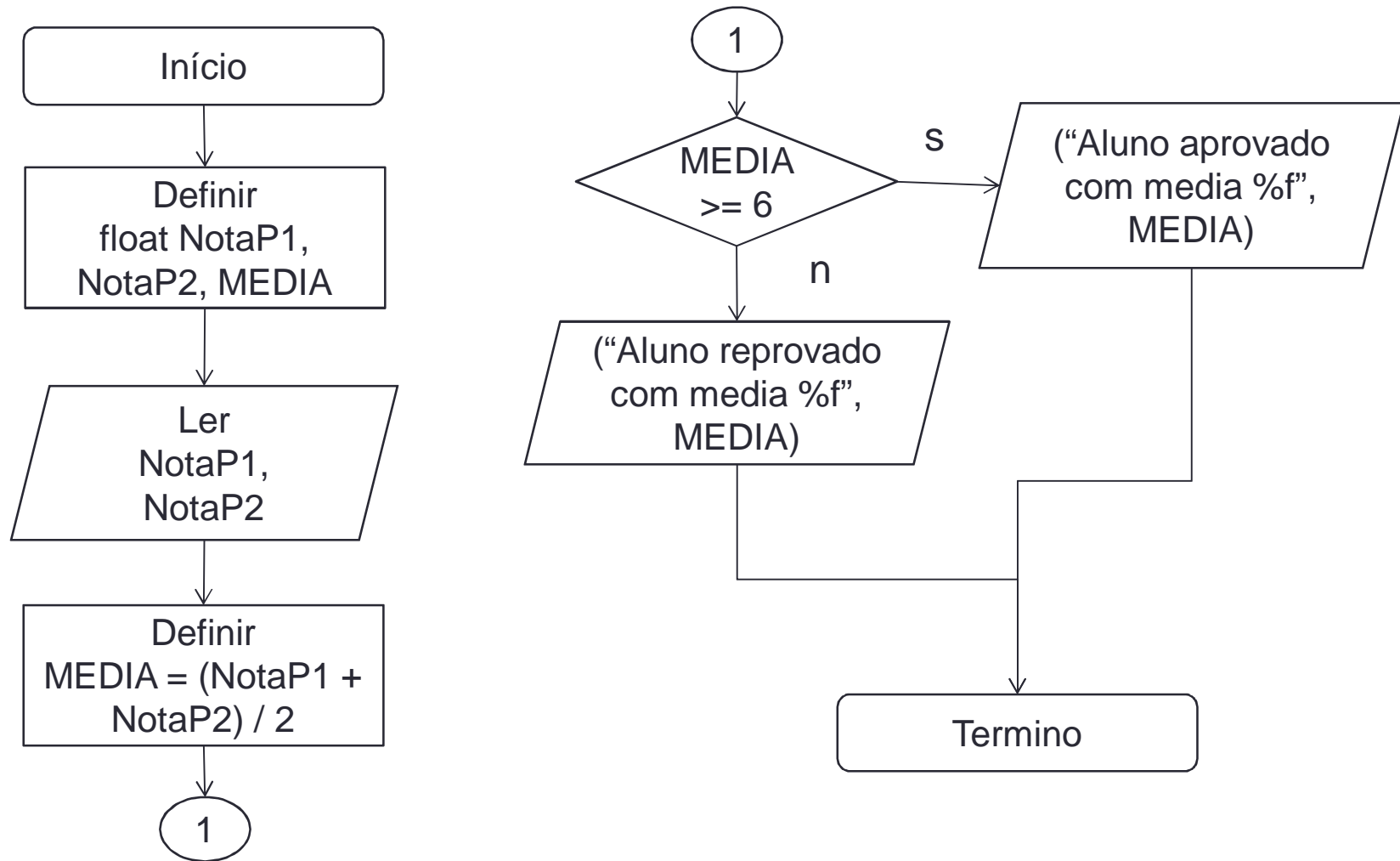


Para exercícios simples, não é necessário.

Mas para exercícios complexos, é recomendável fazer um fluxograma para organizar a lógica, antes de começar a programar.

A programação não deve ser feita por tentativa e erro. Você deve planejar antes o código, para que ele faça exatamente o que você quer.

# Exemplo de fluxograma



# Atividade Extra

- Use a internet, e pesquise:
  - a) O que são linguagens compiladas e interpretadas
  - b) Quais as linguagens de programação mais utilizadas?
  - c) Quais as vantagens e desvantagens da linguagem C



# Aula 2

## Objetivos

- Entender tipos de dados
- Entender variáveis e constantes
- Entender operações e expressões

# Programa adição

```
/* 1.2.Adicao
```

```
    Permite que o usuario entre com dois numeros e exhibe a soma
```

```
*/
```

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

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    int A, B, X;
```

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

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

```
    X = A + B;
```

```
    printf("%d\n", X);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Digite o programa ao lado e execute no Dev C++

**Não se preocupe em entender os comandos, isso será feito nos próximos slides.**

# Tipos de Dados

Dados são elementos “brutos”. São a unidade básica processadas pelos computadores.

Podem ser números (inteiros ou reais), caracteres e lógicos.

## Tipos de Dados Inteiros

Tipo de Dado Inteiro	Faixa de abrangência
int	De -32.768 a 32.768
long int	De -2.147.483.648 a 2.147.483.647
unsigned int	De 0 até 65.535
unsigned long int	De 0 até 4.292.967.295

# Tipos de Dados

## Tipos de Dados Reais

Tipo de Dado Real	Faixa de abrangência
Float	De $3.4 \times 10^{-38}$ até $3.4 \times 10^{+38}$
double	De $1.7 \times 10^{-308}$ até $1.7 \times 10^{+308}$
long double	De $3.4 \times 10^{-4932}$ até $1.1 \times 10^{+4932}$

## Tipos de Dados Caracter

Contém letras, números e símbolos especiais, delimitados entre aspas

## Tipos de Dados Lógicos

Dado lógico ou booleano correspondem a 0 (falso) ou 1 (verdadeiro).

# Variáveis

Variável: é uma área da memória previamente identificada, que tem por objetivo armazenar dados por um determinado tempo. É necessário identificar o tipo de dado que será guardado na variável. Ela sempre é definida com um nome, para fazer referência a variável depois no programa.

Ex de sintaxe de definição de variável:

```
int A;
```

```
char RESPOSTA;
```

```
float NOTA;
```

```
int I, J = 0, K;
```

Na linha acima, são definidas 3 variáveis, do tipo inteiro, sendo que a variável J é inicializada com 0.

# Variável

Algumas regras sobre o nome da variável:

- Podem ser atribuídos com um ou mais caracteres;
- O primeiro caractere do nome de uma variável não pode ser número, deve ser sempre uma letra;
- Não pode conter espaços em branco
- Não podem ser usados como nome de variável palavras reservadas, tais como os nomes de comandos
- Além de letras e números, pode ser usado o underscore  
ex: NOME\_ALUNO
- A linguagem C é case sensitive, ou seja, há diferença entre caracteres maiúsculos e minúsculos

# Contantes

- Reservam uma área da memória e armazena um conteúdo fixo. É diferente da variável, que permite alterar o conteúdo durante a execução do programa.
- Exemplo de definição de constante:

```
#define VALOR_PI 3.14159
```

Neste exemplo, foi definida uma constante VALOR\_PI, que poderá ser utilizada ao longo do programa.

# Operadores Aritméticos

- Usados nos cálculos matemáticos.

Operador	Operação	Resultado
+	Adição	Inteiro ou Real
-	Subtração	Inteiro ou Real
*	Multiplicação	Inteiro ou Real
/	Divisão	Inteiro ou Real
%	Resto da divisão	Inteiro
pow(base,expoente)	Exponenciação	Real
sqrt(valor número)	Raiz quadrada	Real



# Expressões aritméticas

Chaves e colchetes são abolidos. Usa-se apenas parênteses. O computador resolve primeiro os parênteses mais internos da expressão, indo sucessivamente para os mais externos. Exemplo:

$$x = (43 * (55 / (3 + 2)))$$

Executa primeiro  $3 + 2$

Depois  $55 / 5$

Depois  $43 * 11$

Resultado = 473

# Observações

- Por se tratar **main()** de uma função, ela devolve um valor ao sistema operacional. Na maioria dos compiladores de linguagem C é 0. Utiliza-se portanto `return(0)` para forçar o retorno do valor como zero e cumprir exigências do padrão ANSI C.
- A função `main()` pode ser declarada com parâmetros. Mas na maioria dos casos, estes não são declarados. Ou seja, na maioria dos casos usa-se

`int main (void)`

- Void significa um tipo indefinido.

# Programa adição

Neste momento, você já está apto a entender melhor o programa 1.2.Adicao

```
/* 1.2.Adicao
```

```
    Permite que o usuario entre com dois numeros e exhibe a soma
```

```
*/
```

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

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    int A, B, X;
```

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

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

```
    X = A + B;
```

```
    printf("%d\n", X);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Pesquise:

a) O que faz o comando scanf?

Sintaxe:

scanf("expressão de controle", lista de argumentos)

A expressão de controle é um código de formatação para o tipo de dado a ser processado, precedidos pelo sinal %.

# Expressões de controle

No comando scanf, a expressão de controle é um código de formatação para o tipo de dado a ser processado, precedidos pelo sinal %.

Código	Função
%c	Permite seja feita a leitura ou escrita de um caractere
%d	Números inteiros decimais
%e	Números em notação científica
%f	Números reais (ponto flutuante)
%s	Série de caracteres, também chamado de string
%u	Número decimal sem sinal
%x	Número hexadecimal

# Formatação

- A mesma tabela do slide anterior serve como referência para formatar a exibição de dados de variáveis com o comando `printf()`
- Cabe observar que no caso do `%f` é possível indicar a formatação da seguinte forma: `%A.Bf`, onde A é o número de dígitos inteiros e B o número de casas decimais. Ex: `printf("%5.2f",MEDIA)`, formatará a exibição com 5 inteiros e duas casas decimais.
- Existem também outros códigos especiais de formatação:

Cód.	Finalidade
<code>\n</code>	Gera uma nova linha
<code>\t</code>	Gera um espaço de tabulação
<code>\b</code>	Executa um retrocesso de espaço
<code>\"</code>	Apresenta o símbolo aspas
<code>\\</code>	Apresenta o símbolo barra
<code>\f</code>	Gera um salto de página (quando a saída é uma impressora)
<code>\0</code>	Gera um caracter nulo

# Atividade

Programa: 1.3.Salario

- Elaborar um programa que peça para o usuário digitar quantas horas de trabalho (HT), o valor da hora (VH), e o percentual de desconto (PD).
- O programa deve exibir o salário bruto ( $HT * VH$ )
- O valor do desconto em \$
- O salário líquido

**Caso ainda não tenha terminado, finalize primeiro o fluxograma deste programa.**

# Exercícios Extras

Programa: 1.4.AreaRetangulo

- Pedir para o usuário digitar os valores dos lados de um retângulo.
- Exibir Area e Perímetro

---- lay-out desejado

Digite os valores dos lados de um retângulo:

Lado A: \_

Lado B: \_

Area do Retangulo: (exibir o valor)

Perímetro: (exibir o valor)

**Antes de fazer o programa, elabore o fluxograma.**

# Exercícios Extras

Programa: 1.5.AreaCirculo

- Pedir para o usuário digitar o valor do raio
- Exibir Area e Perímetro

---- lay-out desejado

Digite os valor do raio do circulo:

Area: (exibir o valor)

Perímetro: (exibir o valor)

Usar a função pow slide 7

Adicionar no início do arquivo `#include <math.h>`

Que é a biblioteca com as funções matemáticas

Defina uma constante para PI

**Antes de fazer o programa, elabore o fluxograma.**