

# Disciplina de Algoritmos e Programação

# Aula Passada

# 1º Programa: olamundo.c

```
#include <stdio.h> // biblioteca padrão de E/S  
  
/*  
Programa Olá Mundo  
*/  
  
int main(void) {  
    printf("Olá mundo\n"); //exibe Olá mundo  
    return 0; //retorna 0 indicando sucesso  
}
```

Comentários

Função principal

Bibliotecas

Valores de retorno

Comando de saída (biblioteca)

Constante string

# Prática

- Entendendo o ambiente
  - Linux, Kate, GCC, etc...

# Aula de Hoje (teórica)

- Conceito de variável
  - Diferenciação entre valores que permanecem constantes em um problema e de valores que variam
  - Regras para o estabelecimento de nomes de variáveis
  - Processo de transferência de dados da memória para a CPU
  - Tipos de variáveis (int, float, double, char, void)
    - Modificadores de tipo (long, unsigned, cast)
  - Compatibilidade entre tipos

# Identificação dos Dados

- Dados de Entrada e Saída do Problema
  - Números
  - Caracteres
- Manipulação dos dados
  - Processador x Memória
    - Acesso/Modificação dos dados
  - Operações
    - Aritméticas
    - Lógicas

# Conceito de Variável

- Basicamente, uma variável é uma referência a uma posição na memória do computador
  - Analogia:
    - Estante com diversas caixinhas numeradas de forma crescente (Memória do Computador)
    - A declaração de uma variável cria um identificador para uma caixinha específica (Nome da variável)
    - Toda atribuição de valor a este identificador corresponde ao armazenamento do valor na caixinha

# Conceito de Variável

- Basicamente, uma variável é uma referência a uma posição na memória do computador
  - Analogia:
    - Portanto, ...
      - vamos precisar PELO MENOS de um número de caixinhas proporcional a quantidade de variáveis diferentes identificadas no enunciado de um problema
      - POSSIVELMENTE precisemos de caixinhas auxiliares para conseguir fazer o processamento dos valores de entrada para , então, apresentar a saída necessária



# Declaração de Variáveis

- Sintaxe: **tipo lista\_variaveis;**
  - **tipo**: deve ser um **tipo de dado** válido
  - **lista\_variaveis**: um ou mais **identificadores**  
separados por vírgula

# Tipos de Dado

- Tipos Básicos

Tipo	Bytes (8 bits)	Faixa de Valor	Descrição
char	1	-128 à 127	Caractere
int	2	-32768 à 32767	Inteiro
float	4	-3.4E-38 à 3.4E+38	Reais (fracionários)
double	8	-1.7E-308 à +1.7E+308	Reais (fracionários)
void	0	Sem valor	Sem valor (funções)

# Tipos de Dado

- Modificadores de Tipo

Modificador	Modificação	Descrição
signed	c/sinal	Números positivos e negativos
unsigned	s/sinal	Números positivos
long	Longo	Aumenta número de bytes do tipo
short	Curto	Diminui número de bytes do tipo

# Declaração de Variáveis

```
int v1, v2;
```

**tipo** int - inteiro

**lista\_variaveis** - identificadas por v1 e v2

- Ou seja, duas variáveis que podem ser usadas para armazenar valores inteiros com sinal pertencentes a faixa [-32768, ..., +32767]

# Identificadores Válidos

- Primeiro caractere deve ser letra ou sublinha
- Caracteres seguintes devem ser letras, números ou sublinha
- Observações:
  - Letras minúsculas e maiúsculas são tratadas como caracteres diferentes
  - Palavras reservadas: auto, asm, ..., break, char, ..., const, ..., do, else, ...
    - Palavras que já tem definição dentro da linguagem

# Inicialização de Variáveis

- Pode ser feita junto da declaração ou após ela
  - Ex. junto à declaração

```
int variavel = 1; // declaração e inicialização da variável inteira
```

- Ex. após a declaração

```
int variavel; // declaração da variável inteira
```

```
variavel = 1; // inicialização da variável inteira
```

- A inicialização está ocorrendo com a constante 1.  
Veremos que esta inicialização pode ser feita de outras formas.

# Tipo de Variáveis

- Observações
  - Uma variável pode ser interpretada como sendo de outro tipo usando “cast”

Ex.: Variável float interpretada como inteira

```
float var_real = 1.5;
```

```
(int)var_real // somente a parte inteira é considerada
```

# Programa exemplo: data.c

- Escreva um programa com três variáveis – dia, mês e ano – que exiba na tela o conteúdo destas, inicializadas pelo programador, no formato DD/MM/AAAA
  - A) Programa com inicialização junto à declaração
  - B) Programa com declaração e inicialização posterior
  - C) Programa com declaração de múltiplas variáveis de mesmo tipo na mesma linha



# Programa exemplo: data.c

- Escreva um programa com três variáveis – dia, mês e ano – que exiba na tela o conteúdo destas, inicializadas pelo programador, no formato DD/MM/AAAA
  - A) Programa com inicialização junto à declaração
  - B) Programa com declaração e inicialização posterior
  - C) Programa com declaração de múltiplas variáveis de mesmo tipo na mesma linha
    - Detalhe: Como fazer a exibição das variáveis na tela?
      - » Necessário reservar espaço para o conteúdo das variáveis na constante string

# Exercícios

1. Calcular a área de um triângulo. Recebe o valor da base. Recebe o valor da altura. Calcula a área. Mostra a área.
2. Descobrir quais os números adjacentes. Recebe um valor digitado pelo usuário. Mostra o seu sucessor. Mostra o seu antecessor.
3. Mostrar o quadrado de um número. Recebe o número. Calcula seu quadrado. Mostra seu quadrado.
4. Mostrar o cubo de um número. Recebe o número. Calcula seu cubo. Mostra seu cubo.
5. Cálculo percentual. Recebe o peso de uma pessoa. Recebe o percentual de emagrecimento. Calcula o novo peso. Mostra seu novo peso.
6. Conversão Celsius em Fahrenheit. Recebe a temperatura em Celsius. Converte a temperatura em Fahrenheit ( $\text{celsius} * 1.8 + 32$ ). Mostra a Temperatura convertida.
7. Em quantos meses o custo da transição para energia renovável será compensado? Recebe o Custo de implementação. Recebe o Custo mensal energia convencional. Calcula a quantidade de meses. Mostra a quantidade de meses.