ANHANGUERA EDUCACIONAL

Programação Estruturada I

Prof. Ivair Teixeira

labhardware.fav@unianhanguera.edu.br
http://ivairt.vila.bol.com.br

Nessa aula:

- Compilando em C
- Função main()
- Bibliotecas / printf();
- Constantes e Variáveis

Bibliografia:

TLC – Mizrahi, Victorine V. – Treinamento em linguagem C
CCT - Schildt, Herbert – C completo e total

Compilando em C

Compilar um programa consiste em três passos:

- **Criar** o programa (código fonte – nome.c).
- **Compilar** o programa (código objeto – nome.o)
- "**Linkeditar**" o programa (código executável – nome.exe).

O **CodeBlocks** permite realizar as três etapas com o mesmo programa.

Função main()

Um programa em C consiste em uma ou várias "funções".

```
main()
{
   . . .
   . . .
   . . .
}
```

Tipo e nome da função

Marcador de início da função

Local dos comandos

Marcador de fim da função

Todo programa em C deve ter a função main()

Função main()

Estrutura mínima de um programa C

```
main()
{



}
```

Entre as chaves são escritas as instruções

Instruções

- São as comandos ou ações do programa que serão executadas na **ordem** em que foram escritas.
- Devem estar **entre** as chaves.
- São **encerradas** por um "ponto-e-vírgula".
- Podem ser **outra** função, definida em uma biblioteca.
- Sempre utilizar caracteres minúsculos para os nomes (main() é diferente de Main()).

Bibliotecas

•Os comandos da linguagem C não oferecem todas as funcionalidades necessárias para a elaboração de um programa, por exemplo: **entrada e saída**.

•Portanto, todo compilador vem com uma biblioteca de **funções padrão** que realizam as ações mais comuns, complementando as instruções necessárias.

•As bibliotecas devem ser declaradas no **início** do código, antes da função main().

•#include <stdio.h>

A função printf();

•É uma das funções de **E/S** (entrada e saída) usada em C.

•Entre os parênteses está a informação que deve ser **mostrada na tela**.

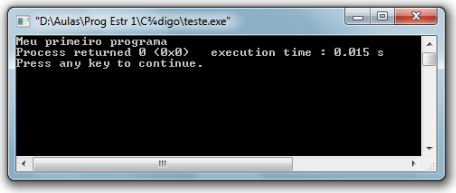
•Está na biblioteca <stdio.h>

•Uma das suas formas é:

printf("mensagem");

A função printf();

```
#include <stdio.h>
main()
{
   printf("Meu primeiro programa");
}
```

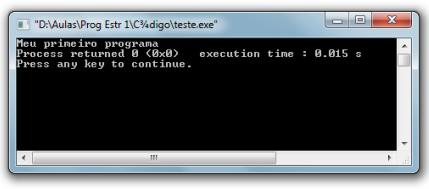


A função printf();

```
#include <stdio.h>
main()
{
   printf("Meu ");
   printf("primeiro ");
   printf("programa");
}
```


A função printf();

```
#include <stdio.h>
main()
{
   printf("Meu ");
   printf("primeiro ");
   printf("programa");
}
```



A função printf();

•Códigos especiais

•O \n é um código que provoca a mudança para uma nova linha. Desempenha a mesma ação do [enter].

printf("Faculdade\n Anhanguera \nde Valinhos");

A função printf();

- Códigos de formatação.
- printf = **print f**ormated
- Outra forma de printf() é:

printf("texto e controle", argumentos);

Por ex:

printf("A raiz quadrada de 81 é: %d", 9)

- Texto = Caracteres gerais (letras e números).
  - Controle = Códigos de formatação.
  - Argumentos = Valores e variáveis.

A função printf();

```
#include <stdio.h>
main()
{
   printf("%s tem %d anos de idade","Paulo",25);
}
```

A função printf();

- Códigos especiais

\n	**Nova linha**
\r	Retorno do cursor
\t	TAB (8 espaços)
\b	Retrocesso
\"	Aspas
\\	Barra
\0	Nulo
\a	Gera um beep

- Códigos de Formatação

%c	Caractere
%d	**Decimal**
%e	Notação Científica
%f	**Ponto Flutuante**
%o	Octal
%s	**Cadeia de Caracteres**
%u	Decimal sem sinal
%x	Hexadecimal
%ld	Decimal Longo
%lf	Ponto Flutuante Longo

Comentários

- Para um perfeito entendimento de um programa é muito importante a utilização de comentários.
- Comentário de uma linha - **//** texto...
  - Todo o texto após **//** é ignorado pelo compilador.
- Comentário de várias linhas - **/*** texto ***/**
  - Todo o texto entre **/*** e ***/** é ignorado pelo compilador

```
#include <stdio.h>
main()
{
   //Função que imprime uma mensagem na tela
   printf("Meu primeiro programa");
}
```


Constantes

- Uma constante tem valor fixo e inalterável em tempo de execução.
- Exemplos:
  - **'c'** - Constante do tipo caractere.
  - **"Anhanguera"** - Const. tipo cadeia de caracteres.
  - **8** - Constante do tipo numérica.

printf("Este é o número: %d", 2);

constante

Variáveis

- Uma variável é um espaço na memória reservada para armazenar um certo tipo de dado.
- Toda variável tem um nome para referenciar seu conteúdo

```
#include <stdio.h>
main()
{
   int num;
   num = 2;
   printf("Este é o número: %d", num);
}
```

Variáveis – Declaração.

•Uma declaração de variável é uma instrução para:

"Reservar uma quantidade de memória apropriada para armazenar um tipo específico de dado".

Por exemplo:

```
int idade;
```

•Reserva um espaço para armazenar um número decimal e esse espaço será referenciado pela palavra "idade".

•Pode-se declarar mais de uma variável do mesmo tipo.

Por exemplo:

```
int idade, rg, cpf;
```


Variáveis – Tipo.

•O **tipo** de uma variável informa a quantidade de memória, em bytes, que esta ocupará na memória.

Tipo	Bits	Bytes	Escala
char	8	1	-128 a 128
int	16/32	2/4	-32768 a 32767
float	32	4	3.4e-38 a 3.4e+38
double	64	8	1.7e-308 a 1.7e+308
void	0	0	Sem valor

Modificadores: long ou long int (4 bytes) unsigned long
unsigned char (0 → 255) short (2 bytes)
unsigned int (0→65535)

Variáveis – Tipo.

•**char**: Armazena um único caractere ASCII, seja ele letra, número ou símbolo especial (&,%,$,@, ...).

•**int**: Armazena um número inteiro entre -2147483647 e 2147483647 (CodeBlocks – Windows. Para garantir a compatibilidade entre -32768 e 32768).

•**float**: Armazena um número com casas decimais (ponto flutuante) de 32 bits com precisão de 6 casas decimais

•**double**: Armazena um número com casas decimais (ponto flutuante) de 64 bits com precisão de 10 casas decimais

•**void**: void=vazio, tipo utilizado em funções que não retornam valores.

Variáveis – Tipo.

```
main()
{
   char letra1 = 'a';
   char letra2 = 'b';
   int valor1 = 10;
   int valor2 = 20;
}
```

	Endereço	Conteúdo
letra1	2686791	a
letra2	2686790	b
	2686789	
	2686788	
	2686787	
	2686786	
valor2	2686785	
	2686784	10
	2686783	
	2686782	
valor2	2686781	
	2686780	20
	2686779	

Variáveis – Inicialização.

•Uma variável pode receber um valor no momento da sua inicialização.

•Por Exemplo:

```
•int num1 = 25;          //número inteiro
•char opcao = 's';       //caractere
•float altura = 1.72     //número ponto flutuante
```


Variáveis – Nomes.

•Escolher sempre nomes significativos.
  •0pcao, idade, altura, num1

•Deve ser uma única palavra, sem espaço.

•O primeiro caractere sempre deve ser uma letra ou o caractere de sublinhar "_".
  •valor, _valor, soma.

•Os demais caracteres podem ser letras números ou o caractere de sublinhar.

•Não pode ser uma palavra chave da linguagem.
  •int, char, for, goto,...

Variáveis – Nomes.

•Palavras chaves em C

auto	**double**	**int**	**struct**
break	**else**	**long**	**switch**
case	**enum**	**register**	**typedef**
char	**extern**	**return**	**union**
const	**float**	**short**	**unsigned**
continue	**for**	**signed**	**void**
default	**goto**	**sizeof**	**volatile**
do	**if**	**static**	**while**

•Atenção: C diferencia maiúsculas de minúsculas.
Portanto **nun** é diferente de **Num** e diferente de **NUM**.

Atividades:

1 – O que são instruções?

2 - Para que servem as bibliotecas?

3 – Porque é importante comentar um código fonte? Quais os caracteres especiais usados para esse fim.

4 – Qual a diferença entre constantes e variáveis.

5 – Quais são os tipos básicos de variáveis usados em C (sem os modificadores). Para que serve cada tipo?

6 – Elabore um programa que declare e inicialize uma variável de cada tipo: char, int, float e double.

7 – Elabore um programa de que imprima frase a seguir, usando variáveis para os valores em negrito:

Paulo tem **23** anos, e pesa **60,5** kg

Atividades:

Considere que os códigos estão inseridos corretamente na função main() e as devidas bibliotecas declaradas:

8 – Qual/quais os erros da linha de código abaixo ?

```
Printf( Existem %c alunos nessa sala, 40)
Char op = a;
int num1 = 3.5;
```

9 – Assinale as declarações **NÃO** válidas e explique a razão.

a)( ) int 1_valor =10;
b)( ) char op = c;
c)( ) float num = 100;
d)( ) int INT = 32000;
e)( ) Double num = 54;
f)( ) float num-2 = 30.5;
g)( ) int valor_final = 0;
h)( ) char ____A = 'x';
i)( ) int letra = 8;
j)( ) char Char = "j";
k)( ) float pi = 3.1416;
l)( ) int A1209 = 10000;

Atividades:

•Leiam o capítulo 1 o PLT.

•Respondam os exercício das páginas 24, 25 e 26.