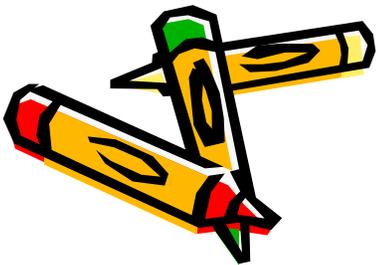


CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS



DISCIPLINA : INTRODUÇÃO À
LÓGICA DE PROGRAMAÇÃO

ASSUNTO: C



PROFESSOR : Romilson Lopes Sampaio

C

Linguagem inventada e implementada primeiramente por Dennis Ritchie utilizando o sistema operacional UNIX.

Características da Linguagem C

- C é uma linguagem compilada – lê todo o código fonte e gera o código objeto (ling. máquina) uma única vez, enquanto que, as Linguagens Interpretadas: lêem o código fonte, traduzem e executam o programa cada vez que for executado.
- Case sensitive – sensível a maiúsculas

Diretivas para o processador - Bibliotecas

- Diretiva `#include` permite incluir uma biblioteca
- Bibliotecas contêm funções pré-definidas, que permitem realizar tarefas específicas nos programas
- As primeiras linhas do programa indicam as bibliotecas utilizadas
- Exemplos

<code>#include <stdio.h></code>	Funções de entrada e saída
<code>#include <stdlib.h></code>	Funções padrão
<code>#include <math.h></code>	Funções matemáticas
<code>#include <string.h></code>	Funções de texto

Estrutura básica de um programa C

Todo programa C inicia sua expressão chamando a função **main()**, sendo obrigatória a sua declaração no programa principal;

```
main () /* Função obrigatória */
{ /*inicio*/
    declaração de variáveis locais da função main
    comandos da função main
} /*fim*/
```

Estrutura básica de um programa C

```
/* Primeiro Programa em C */ comentários

#include <stdio.h> /*biblioteca de E/S */

main()      /*função principal - inicio do
              programa*/
{
    /*marca início da função*/
    printf("Meu primeiro programa em C\n");
    /*escreve no monitor e muda de
linha*/
    getchar();
}           /*marca o fim da função*/
```

Características da linguagem C - Comentários

- Comentários são blocos de programa que servem apenas para ajudar o programador e seus colaboradores no processo de documentação do código fonte;

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

```
main() {
```

```
    //Mensagem de saudação
```

```
    printf("Bem vindo!");
```

```
    printf("\nEstou aprendendo Linguagem C! ");
```

```
    //Esperando o usuário pressionar <Enter> para sair do programa
```

```
    printf("\n\nPressione <Enter> para sair do programa.");
```

```
    getchar();
```

```
}
```

Características da linguagem C - Comentários

- O comentário com barra dupla também pode vir ao final da linha que contenha um comando como podemos ver abaixo:

```
#include <stdio.h>

main() {

    printf("Bem vindo!"); //Mensagem de saudação

}
```

Características da linguagem C - Comentários

- Qualquer trecho de programa entre `/*` e `*/` é considerado um comentário. A grande diferença de tipo de comentário é que podemos comentar blocos de linhas inteiras. Um grande uso desse tipo de comentário está nos cabeçalhos de programa onde identificamos o programa:

```
/* Programa exemplo */
```

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

```
main() {
```

```
    //Código qualquer
```

```
}
```

Declarações

- Declaram as variáveis e seus tipos
- Os nomes das variáveis devem conter apenas letras, dígitos e o símbolo _
- Os principais tipos são: **int**, **float**, **double** e **char**
- Exemplos

```
int n;  
int quantidade_valores;  
float x, y, somaValores;  
char sexo;  
char nome[40];
```

C diferencia letras maiúsculas de minúsculas!

```
int n, N;
```

n é diferente de N!

Tipos Básicos de Dados

- determinar um conjunto de valores e as possíveis operações realizadas sobre os mesmos ;
- informa a quantidade de memória (bytes);

tipo	bytes	escala
char	1	-128 a 127
int	2	-32.768 a 32.767
float	4	$-3,4 \times 10^{38}$ a $-3,4 \times 10^{-38}$ e $3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
double	8	$-1,7 \times 10^{308}$ a $-1,7 \times 10^{-308}$ e $1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$

Outros tipos de variáveis

Long ou Long int	(4 bytes)
Unsigned Char	(0 a 255)
Unsigned int	(0 a 65.535)

Programa Exemplo – Tipos de Dados e Variáveis

```
#include <stdio.h>
main( )
{
    int soma=10;
    float money=2.21;
    char letra= 'A';

    printf ("valor da soma = %d\n", soma);
    printf ("Valor de Money = %f\n", money);
    printf("Valor de Letra = %c\n", letra);
    getchar();

}
```

Operadores Matemáticos

Operador	Exemplo	Comentário
+	$x + y$	Soma x e y
-	$x - y$	Subtrai y de x
*	$x * y$	Multiplica x e y
/	x / y	Divide x por y
%	$x \% y$	Resto da divisão de x por y
++	$x++$	Incrementa em 1 o valor de x
--	$x--$	Decrementa em 1 o valor de x

Operadores de Atribuição

Operador	Exemplo	Comentário
=	$x = y$	Atribui o valor de y a x
+=	$x += y$	Equivale a $x = x + y$
-=	$x -= y$	Equivale a $x = x - y$
*=	$x *= y$	Equivale a $x = x * y$
/=	$x /= y$	Equivale a $x = x / y$
%=	$x \% = y$	Equivale a $x = x \% y$

Operadores Relacionais

Operador	Exemplo	Comentário
==	$x == y$	O conteúdo de x é igual ao de y
!=	$x != y$	O conteúdo de x é diferente do de y
<=	$x <= y$	O conteúdo de x é menor ou igual ao de y
>=	$x >= y$	O conteúdo de x é maior ou igual ao de y
<	$x < y$	O conteúdo de x é menor que o de y
>	$x > y$	O conteúdo de x é maior que o de y

As expressões relacionais em C retornam :

- 1 se verdadeiro e;
- 0 se falso.

Operadores Lógicos

- **&&** (E lógico): retorna verdadeiro se ambos os operandos são verdadeiros e falso nos demais casos.
Exemplo: `if(a>2 && b<3)`.
- **||** (OU lógico): retorna verdadeiro se um ou ambos os operandos são verdadeiros e falso se ambos são falsos.
Exemplo: `if(a>1 || b<2)`.
- **!** (NÃO lógico): usada com apenas um operando. Retorna verdadeiro se o operando é falso e vice-versa.
Exemplo: `if(!var)`.

Funções Matemáticas

Função	Exemplo	Comentário
<code>ceil</code>	<code>ceil(x)</code>	Arredonda o número real para cima; <code>ceil(3.2)</code> é 4
<code>cos</code>	<code>cos(x)</code>	Cosseno de x (x em radianos)
<code>exp</code>	<code>exp(x)</code>	e elevado à potencia x
<code>fabs</code>	<code>fabs(x)</code>	Valor absoluto de x
<code>floor</code>	<code>floor(x)</code>	Arredonda o número real para baixo; <code>floor(3.2)</code> é 3
<code>log</code>	<code>log(x)</code>	Logaritmo natural de x
<code>log10</code>	<code>log10(x)</code>	Logaritmo decimal de x
<code>pow</code>	<code>pow(x, y)</code>	Calcula x elevado à potência y
<code>sin</code>	<code>sin(x)</code>	Seno de x
<code>sqrt</code>	<code>sqrt(x)</code>	Raiz quadrada de x
<code>tan</code>	<code>tan(x)</code>	Tangente de x

```
#include <math.h>
```

Comando de atribuição

- Atribui o valor da direita à variável da esquerda
- O valor pode ser:
 - uma *constante*,
 - uma variável ou
 - uma expressão
- Exemplos

```
x = 4;    --> lemos x recebe 4
```

```
y = x + 2;
```

```
y = y + 4;
```

```
valor = 2.5;
```

```
sexo = 'F'
```

Entrada de Dados

- Função **scanf**

scanf(Expression de controle, Lista de variáveis);

Expressão de controle - escrita entre aspas e indica como o sistema deve armazenar os dados digitados no teclado.

Lista de variáveis - variáveis precedidas do *operador de endereço* & (indica o endereço da posição de memória da variável).

Exemplos:

```
int i, j;  
float x;  
char c;  
scanf("%d", &i);  
scanf("%d %f", &j, &x);  
scanf("%c", &c);  
scanf("%s", nome);
```

%d	inteiro
%f	float
%lf	double
%c	char
%s	palavra

Entrada de Dados

Algoritmo Teste

n1, n2, n3, media: Real

leia n1

leia n2

leia n3

leia n1, n2, n3

Na Linguagem C...

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

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

```
main()
```

```
{
```

```
double n1, n2, n3, media;
```

```
scanf ("%lf",&n1);
```

```
scanf ("%lf",&n2);
```

```
scanf ("%lf",&n3);
```

```
scanf ("%lf %lf %lf",&n1, &n2, &n3);
```

```
system("PAUSE");
```

```
}
```

Entrada de Dados

Algoritmo

n1, n2, n3, media: Real

leia n1, n2, n3

media \leftarrow (n1+n2+n3)/3

Na Linguagem C...

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

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

```
main()
```

```
{
```

```
double n1, n2, n3, media;
```

```
scanf ("%lf %lf %lf",&n1, &n2, &n3);
```

```
media=(n1+n2+n3)/3;
```

```
system("PAUSE");
```

```
}
```

Saída de Dados

- Função **printf**

`printf`(Expressão de controle, Lista de argumentos);

Expressão de controle - mensagens a ser exibidas, códigos de formatação idênticos aos códigos de conversão da função `scanf()`, indicando como o conteúdo de uma variável deve ser exibido.

Lista de argumentos - identificadores de variáveis, expressões aritméticas ou lógicas e valores constantes.

Exemplos:

```
int i, j;
float x;
char c;
printf("%d", i);
printf("%d, %f", j, x);
printf("%c", c);
printf("%s", nome);
```

<code>%d</code>	inteiro
<code>%f</code>	float
<code>%lf</code>	double
<code>%c</code>	char
<code>%s</code>	palavra

Saída de Dados

- *Códigos especiais da função printf()*

Código	Ação
\n	leva o cursor para a próxima linha
\t	executa uma tabulação
\b	executa um retrocesso
\f	leva o cursor para a próxima página
\a	emite um sinal sonoro (<i>beep</i>)
\"	exibe o caractere "
\\	exibe o caractere \
\%	exibe o caractere %

Saída de Dados

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int i, j;
    float x;
    char c;
    char nome [10];
    i = 1; j = 2; x = 3;
    printf("%d", i);
    printf("%d, %f", j, x);

    system("PAUSE");
}
```

Saída de Dados

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    double n1, n2, n3, media;
    // entrada de dados
    scanf ("%lf %lf %lf",&n1, &n2, &n3);
    // operacao
    media=(n1+n2+n3)/3;
    // saida de dados
    printf("%f", n1);
    printf("%f", n2);
    printf("%f", n3);
    printf("%f", media);

    system("PAUSE");

}
```

Saída de Dados

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de
    // variaveis
    int i, j;
    float x;
    char c;
    char nome[30];
```

```
//entrada de dados
scanf("%d", &i);
scanf("%d %f", &j,
&x);
scanf("%c", &c);
scanf("%s", nome);
// exibicao de dados
printf("%d", i);
printf("%d, %f", j,
x);
printf("%c", c);
printf("%s", nome);

system("PAUSE");
}
```

Exemplo

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    //declaracao de variaveis
    int n;
    int quantidade_valores;
    float x, y, somaValores;
    char sexo;
    char nome[40];
    // operacao
    somaValores =1+1;
    //saida de dados
    printf("somaValores=%f", somaValores);

system("PAUSE");
}
```

Exemplo

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    int n;
    int quantidade_valores;
    float x, y, somaValores;
    char sexo;
    char nome[40];
    x = 1; y = 2;
    //atribuicao
    somaValores = x + y;
    // mostra mensagem de texto na tela
    printf("alo mundo");
    // mostra valor na tela
    printf("somaValores=%f",
    somaValores);
    system("PAUSE");
}
```

Entrada e Saída

Exemplo

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;
    scanf ("%lf %lf %lf",&n1, &n2,
&n3);
    media=(n1+n2+n3)/3;
    printf ("%lf",media);

    system("PAUSE");
}
```

Exemplo

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;
    printf("Digite 3 notas: ");
    scanf ("%lf %lf %lf",&n1, &n2, &n3);
    media=(n1+n2+n3)/3;
    printf ("Media: %.2lf",media);

    system("PAUSE");
}
```

Exemplo 1

- 1) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando a seguinte fórmula: $\text{peso ideal} = (72.7 * h) - 58$

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{
    float altura, pesoIdeal;
    char sexo;
    printf ("\nDigite sua altura em metros: ");
    scanf ("%f",&altura);
    printf ("\nDigite seu sexo (F/M): ");
    sexo=getche();
    pesoIdeal=(72.7*altura)-58;
    printf ("\nO sexo digitado foi %c e o peso ideal desta pessoa eh %.2f quilos\n",
        sexo, pesoIdeal);
    system("PAUSE");
}
```

Exemplo 2

2) Construa um programa que tendo como entrada dois pontos quaisquer do plano P(x1,y1) e Q(x2,y2), imprima a distância entre eles. A fórmula da distância é:

$$\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{
    float x1, y1, x2, y2;
    float distancia;
    printf ("\nDigite o valor de x1: ");
    scanf ("%f",&x1);
    printf ("\nDigite o valor de y1: ");
    scanf ("%f",&y1);
    printf ("\nDigite o valor de x2: ");
    scanf ("%f",&x2);
    printf ("\nDigite o valor de y2: ");
    scanf ("%f",&y2);
    distancia=sqrt(pow((x2-x1),2)+pow((y2-y1),2));
    printf ("\nA distancia entre os pontos P1 e P2 eh %.2f\n", distancia);
    system("PAUSE");
}
```

Exemplo getchar

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

int main()
{
    char letra;
    printf("Insira um caractere: ");
    letra = getchar();
    printf("Você digitou: '%c'", letra);
    system("PAUSE");
}
```

Exercício

3) Construa um programa que calcule a quantidade de latas de tinta necessárias e o custo para pintar tanques cilíndricos de combustível, onde são fornecidos a altura e o raio desse cilindro.

Sabendo que:

- a lata de tinta custa R\$20,00
- cada lata contém 5 litros
- cada litro de tinta pinta 3 metros quadrados.

Sabendo que:

Área do cilindro= $3.14 * \text{raio}^2 + 2 * 3.14 * \text{raio} * \text{altura}$
e que raio e altura são dados de entrada.

Resposta do Exercício

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{
    float altura, raio, areaCilindro, qtdadeLitros, qtdadeLatas, custo;
    printf ("\nDigite o valor da altura em metros: ");
    scanf ("%f",&altura);
    printf ("\nDigite o valor do raio em metros: ");
    scanf ("%f",&raio);
    areaCilindro=3.14*raio*raio + 2*3.14*raio*altura;
    printf ("\nA area do cilindro eh %.2f metros quadrados", areaCilindro);
    qtdadeLitros=areaCilindro/3;
    printf ("\nA qtidade de litros necessaria eh de %.2f ", qtdadeLitros);
    qtdadeLatas=qtdadeLitros/5;
    printf ("\nA qtidade de latas necessaria eh de %.2f ", qtdadeLatas);
    custo=qtdadeLatas*20;
    printf ("\nO valor total das tintas eh de R$ %.2f \n", custo);
    system("PAUSE");
}
```