



# Linguagem C - Estrutura Sequencial

---

Curso: Ciência da Computação  
Disciplina: Algoritmos e Programação  
Profº Luciano Antunes



# Introdução

---

- A linguagem C foi inventada por Dennis Ritchie e implementada no sistema operacional UNIX na década de 1970;
- Ela teve influências da linguagem B;
- A linguagem C++ é uma extensão da linguagem C que dá suporte a programação à orientação a objetos;



# Introdução

---

- A sintaxe da linguagem C++ é muito similar a da linguagem C;
- A linguagem Java, uma extensão da “família C”, incorpora a orientação a objetos e seu uso em multiplataformas (1995)



# Linguagem C – palavras reservadas

- Nomes utilizados pelo compilador para representar comandos de controle do programa, operadores, etc

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

- Case sensitive.



# Forma Geral de um programa em C

---

```
#include <nome da biblioteca>  
void main()  
{  
    <declaração de variaveis>;  
    <comando 1>;  
    <comando 2>;  
    .....  
    <comando n>;  
}
```



## Nome da Biblioteca

---

Nome da biblioteca: São arquivos contendo várias funções que podem ser incorporadas aos programas escritos em C . A diretiva `#include` faz com que o texto contido dentro da biblioteca especificada seja inserido no programa.



## Algumas bibliotecas do C

Arquivo	Descrição
stdio.h	Funções de entrada e saída (I/O)
string.h	Funções de tratamento de strings
math.h	Funções matemáticas
ctype.h	Funções de teste e tratamento de caracteres
stdlib.h	Funções de uso genérico
locale.h	Acentua as palavras na linguagem C para isso deve-se utilizar o comando: setlocale(LC_ALL,"Portuguese"); Após o main();



## A função main

---

`void main( )` - A função main deve existir em algum lugar de seu programa e marca o ponto de início da execução do programa. Se um programa for constituído de uma única função, esta será `main( )`.

Exemplo:

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    printf("E preciso fazer todos os algoritmos para aprender \n ");
```

```
    getch();
```

```
}
```





# Declaração de variáveis

---

- As variáveis em C , contém os dados sobre os quais seu programa operará.
- São declaradas após a especificação de seus tipos.
- Devem ser declaradas no início do programa.

Os principais tipos básicos que podem ser utilizados na declaração de variáveis são: int, float, double e char;

```
float x;          x=5.5;
```

```
int a;            a=10;
```

```
double b;         b=98.3;
```

```
char sexo;        sexo='f';
```

```
char nome[40];    nome="Carlos";
```

# Identificadores



---

- São utilizados para dar nomes a constantes, variáveis, funções e vários objetos definidos pelo usuário.
- As regras para formação dos identificadores são:
  1. Todo identificador deve iniciar por uma letra (a..z ou A..Z) ou um sublinhado (\_);
  2. Após o primeiro caracter pode ser utilizado: letras, sublinhados e/ou dígitos;
  3. Não pode conter símbolos especiais;
  4. Utiliza-se identificadores de, no máximo, 32 caracteres por estes serem significativos.
  5. Não pode ser palavra reservada e nem nome de funções de bibliotecas.

**Obs: letras maiúsculas e minúsculas são tratadas de forma diferente.**



## Modificadores de Tipo

- Os modificadores de tipo servem para definir as características da variável que está sendo declarada quanto ao seu tamanho em bytes e forma de representação dos valores (inteiros sem sinal ou com sinal, por exemplo).
- Quanto ao sinal:

Modificador	Descrição
signed	A variável pode receber valores negativos ou positivos. Padrão.
unsigned	A variável somente pode receber valores positivos.



# Modificadores de Tipo

- Quanto ao tamanho:

Modificador	Descrição
short	Formato curto (2 bytes). Padrão para compiladores de 16 bits.
long	Formato longo (4 bytes). Padrão para compiladores de 32 bits.

- O tipo padrão (default) em C é int, por isso sempre que a declaração tiver apenas modificadores é assumido o tipo int. Por exemplo, long é equivalente a signed long int. Toda a função sem tipo declarado é assumida como sendo int.



# Expressões

---

- Seu conceito está intimamente ligado ao conceito de expressão (ou fórmula) matemática, onde um conjunto de variáveis e constantes numéricas relaciona-se por meio de operadores compondo uma fórmula que, uma vez validada, resulta num valor. Dividem-se em:
- Expressões Aritméticas;
- Expressões Relacionais;
- Expressões Lógicas.



## Expressões Aritméticas

- São aquelas cujo resultado da expressão é do tipo numérico, seja ele inteiro ou real.

Operador	Exemplo	Comentário
+	$z=x+y$	Soma o conteúdo de x e de y
-	$z=x-y$	Subtrai o conteúdo de y do conteúdo de x
*	$z=x*y$	Multiplica o conteúdo de x pelo conteúdo de y
/	$z=x/y$	Obtém o quociente da divisão de x por y
%	$z=x\%y$	Obtém o resto da divisão de x por y
++	$z++$	Aumenta o conteúdo de z em 1 unidade, semelhante a $z = z + 1$
--	$z--$	Diminui o conteúdo de z em 1 unidade, semelhante a $z = z - 1$

Obs: O operador % só pode ser utilizado com operandos do tipo inteiro. 14



# Expressões Relacionais

- Uma expressão relacional é uma comparação realizada entre dois valores de um mesmo tipo, e o resultado, um valor lógico.

Operador	Matemática	Como utilizar em C
Igual	=	==
Diferente	≠	!=
Maior	>	>
Menor que	<	<
Maior ou igual a	≥	>=
Menor ou igual a	≤	<=



# Expressões Lógicas

- É a expressão cujos operadores são lógicos e cujos operandos são relações, e variáveis do tipo lógico.

Operador	Matemática	Como utilizar em C
Conjunção	E	&&
Disjunção	OU	
Negação	NÃO	!





# Funções Matemáticas

---

- Uma função pode ser definida como sendo a relação onde um conjunto de variáveis e constantes numéricas relaciona-se por meio de operadores, compondo uma fórmula que, uma vez avaliada, resulta num valor. Para a utilização dessas funções, deve-se adicionar ao programa a biblioteca **math.h**



# Funções Matemáticas

Função	Exemplo	Comentário
ceil	ceil(x)	Arredonda um número real para cima, por exemplo, ceil(3.2) é 4
cos	cos(x)	Calcula o cosseno de x (x deve representado em radianos)
exp	exp(x)	Obtém o logaritmo natural e elevado à potência x
fabs	fabs(x)	Obtém o valor absoluto de x
floor	floor(x)	Arredonda um número real para baixo, por exemplo, floor(3.2) é 3
log	log(x)	Obtém o logaritmo natural de x
log10	log10(x)	Obtém o logaritmo de base 10 de x
pow	pow(x,y)	Calcula a potência de x elevado a y
sin	sin(x)	Calcula o seno de x (x deve estar representado em radianos)
sqrt	sqrt (x)	Calcula a raiz quadrada de x
cbrt	cbrt (x)	Calcula a raiz cubica de x
tan	tan (x)	Calcula a tangente de x (x deve estar representado em radianos)



# Palavras Reservadas em C

---

São nomes utilizados pelo compilador para representar comandos de controle do programa, operadores e diretivas.



# Palavras Reservadas em C – Comandos Básicos

---

## ■ **Atribuição: =**

É a principal forma de se armazenar um dado em uma variável. Esse comando permite que você forneça um valor a uma variável, onde o tipo desse valor tem de ser compatível com a variável.

identificador=expressão;

Onde:

Identificador : é o nome da variável à qual está sendo atribuído um valor **= é o símbolo de atribuição**

expressão : pode ser uma expressão aritmética, uma expressão lógica ou literal, cuja avaliação (resultado) é atribuída ao identificador (variável) ; finaliza o comando



## Comando de Saída em C

---

- É o comando responsável por enviar um resultado, uma informação, ao usuário.
- O valor de cada variável é pesquisado na memória e inserido em um dispositivo de saída. Através desse comando, o computador pode emitir os resultados e outras mensagens para o usuário através da tela do computador ou uma impressora.



## Comando de Saída em C - printf

---

`printf(formato, lista de constantes/variáveis/expressões...);`

- O primeiro parâmetro é uma cadeia de caracteres, em geral, delimitada, com aspas, que especifica o formato de saída das constantes, variáveis e expressões listadas em seguida.
- Para cada valor que se deseja imprimir, deve existir um especificador de formato correspondente na cadeia de caracteres formato.



# Comando de Saída em C – printf – Especificadores de Formato

Código	Formato
%d	Especifica um decimal
%o	Especifica um octal
%x	Especifica um hexadecimal
%u	Especifica um unsigned int (decimal sem sinal)
%f	Especifica um double (ou float)
%e	Especifica um double (ou float) em formato científico
%g	Especifica um double (ou float) em formato científico no formato mais apropriado (%f ou %e)
%c	Especifica um char
%s	Especifica uma cadeia de caracteres
%%	Especifica o %



# Comando de Saída em C – printf - exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int x;
    x=10;
    printf("Conteudo de x em decimal: %d", x);
    printf("\nConteudo de x em octal: %o", x);
    printf("\nConteudo de x em hexadecimal: %x", x);
    getch();
}
```





# Comando de Saída em C – printf – Especificadores de Formato

Código Especial	Formato
\n	Nova Linha
\t	Tabulação
\"	Aspas
\\	Barra



## Comando de Saída em C – printf – Especificadores de Formato

- Caso seja necessário, é possível também, especificar o tamanho dos campos, como ilustrado a seguir.

`%4d`

		3	3
4			

`%7.2f`

				5	.	3	0
7							



## Comando de Saída em C – printf - exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    printf("A quantidade de funcionários e: %2d", 350);
    printf("\nA quantidade de funcionários e: %4d", 350);
    printf("\nA quantidade de funcionários e: %6d", 350);
    getch();
}
```



# Comando de Saída em C – printf - exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    printf("Valor = %4.2f", 1234.56);
    printf("\nValor = %3.2f", 1234.56);
    printf("\nValor = %3.1f", 1234.56);
    printf("\nValor = %10.3f", 1234.56);
    printf("\nValor = %010.3f", 1234.56);
    printf("\nValor = %-10.3f", 1234.56);
    getch();
}
```



## Comando de Saída em C – printf - exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int prod;
    prod = 28 * 43;
    printf(" O produto entre 28 e 43 é: %d\n",prod);
    getch();
}
```



## Comando de Saída em C – printf - exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float media;
    media = (8+9+7)/3;
    printf(" A media entre 8, 9 e 7 é: %f\n",media);
    getch();
}
```



## Comando de Entrada em C

---

- É o comando que permite que o usuário digite dados, possibilitando um “diálogo com o computador”. O dado digitado é armazenado em uma variável declarada no início do programa.



## Comando de Entrada em C - scanf

---

**scanf(formato, lista de endereços das variáveis);**

- O formato deve possuir especificadores de tipos similares aos mostrados para a função printf. Para a função scanf, no entanto, existem especificadores diferentes para o tipo float e o tipo double.





# Comando de Entrada em C - scanf

Código	Formato
%d	Especifica um int (formato decimal)
%o	Especifica um int (formato octal)
%x	Especifica um int (formato hexadecimal)
%u	Especifica um unsigned int (decimal sem sinal)
%f	Especifica um float
%e	Especifica um float em formato científico
%g	Especifica um float em formato científico no formato mais apropriado
%lf	Especifica um double
%lê	Especifica um double em formato científico
%lg	Especifica um double em formato científico no formato mais apropriado
%c	Especifica um char
%s	Especifica uma cadeia de caracteres



## Comando de Entrada em C - scanf

---

A lista de endereços das variáveis deve consistir nos endereços das variáveis. A linguagem C oferece um operador para tipos básicos chamada **operador de endereço** e referenciado pelo símbolo **&** que resulta o endereço da variável.

Ex:

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

Um valor digitado pelo usuário será armazenado na variável **x**;



# Comando de Entrada em C – scanf - Exemplos

---

## **O operador de endereço &**

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    int num;
```

```
    num = 2;
```

```
    printf("Valor = %d, endereco=%u", num, &num);
```

```
    getch();
```

```
}
```



# Comando de Entrada em C – scanf - Exemplos

---

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int prod, num1, num2;
    printf("Digite o primeiro numero: ");
    scanf("%d",&num1);
    printf("Digite o segundo numero: ");
    scanf("%d",&num2);
    prod = num1 * num2;
    printf(" O produto entre os numeros %d e %d e: %d",num1, num2,
    prod);
    getch();
}
```