

LABORATÓRIO DE PROGRAMAÇÃO

Tipos de Dados e Operadores em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.
Cursos de Computação
1º. Período

Laboratório de Programação

Tipos de Dados e Operadores em C

Variável

- “objeto” que pode assumir diversos valores;
- espaço de memória de um certo tipo de dado associado a um nome para referenciar seu conteúdo

```
#include<stdio.h>
#include<stdlib.h>
int main( )
{
    int idade;
    idade = 30;
    printf (“ A idade é : %d”, idade);
    system(“pause”);
    return (0);
}
```

Laboratório de Programação

Tipos de Dados e Operadores em C

Nomes de Variáveis

- quantos caracteres quiser (32);
- comece com letras ou sublinhado:
Seguidos de letras, números ou sublinhados
- a linguagem C é sensível ao caso:
peso ≠ Peso ≠ pEso ≠ PESO
- não podemos definir um identificador com o mesmo nome que uma palavra chave
auto static extern int long if while do

Laboratório de Programação

Tipos de Dados e Operadores em C

- Uma variável não pode ter o mesmo nome de uma palavra reservada do C e não deverá ter o mesmo nome de uma função.

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

Laboratório de Programação

Tipos de Dados e Operadores em C

Tipos de Variáveis

- O tipo de uma variável informa a quantidade de memória, em bytes, que ela ocupará
 - Informa o modo como um valor deverá ser armazenado;
- Em C, existem cinco tipos básicos de variáveis:

`char`

`int`

`float`

`double`

`void`

Laboratório de Programação

Tipos de Dados e Operadores em C

Tipos de Variáveis

- Com exceção de **void** todos os tipos básicos de dados podem ser acompanhados por um modificador.
- Em C temos 3 modificadores: **short**, **long** e **unsigned**
- Um modificador de tipo pode ser utilizado sem que seja especificado o tipo da variável.
- Quando isso é feito, o compilador assume, por padrão, que o tipo é **int**.

Laboratório de Programação

Tipos de Dados e Operadores em C

Tipos de Variáveis

Tipo	Bits	Bytes	Escala
char	8	1	-128 a 127
int	32	4	-2.147.483.648 a 2.147.483.647
short	16	2	-32.765 a 32.767
long	32	4	-2.147.483.648 a 2.147.483.647
unsigned char	8	1	0 a 255
unsigned	32	4	0 a 4.294.967.295
float	32	4	3.4×10^{-38} a 3.4×10^{38}
double	64	8	1.7×10^{-308} a 1.7×10^{308}
long double	80	10	3.4×10^{-4932} a 3.4×10^{4932}
void	0	0	Nenhum valor

Laboratório de Programação

Operadores Aritméticos

- Representam cálculos matemáticos, operando sobre os tipos integrais¹ e de ponto flutuante²:

Operador	Ação Inteiros	Ação Float
+	Adição	
-	Subtração	
++	Incremento por 1	
--	Decremento por 1	
*	Multiplicação	
/	Divisão com resto	Divisão real
%	Resto de Divisão	Erro de compilação
=	Atribuição	

1. int, short, long, long long, char, etc.
2. float, double long double, etc.

Laboratório de Programação

Operadores Aritméticos

- **Adição e Subtração**
 - Operações binárias.
 - São avaliadas como a soma/subtração dos dois operandos.

Exemplo:

“a + b”

“1 + 1”

“5 – 4”

“50 – 8”

Laboratório de Programação

Operadores Aritméticos

- **Incremento e Decremento**
 - Operações unárias.
 - São avaliadas como o valor do operador e modificam esse operador.
 - Deve ser tomado cuidado com a ordem em que são usados:
 - Se usado prefixado, o operador é modificado **ANTES** da operação ser avaliada.
 - Se usado pós-fixado, o operador é modificado **DEPOIS**.

“++x” será avaliado como “x + 1”, e x será igual a “x + 1”

“x++” será avaliado como “x”, e x será igual a “x + 1”

Laboratório de Programação

Operadores Aritméticos

- **Multiplicação e Divisão**
 - Operações binárias.
 - São avaliadas como a multiplicação/divisão dos operandos.
 - Com tipos integrais, a divisão tem resto.
 - Com tipos de ponto flutuante, não.

Exemplo:

“6/4” (avaliado como “1”)

“3.2/2” (avaliado como “1.6”)

Laboratório de Programação

Operadores Aritméticos

Exemplo:

```
#include<stdio.h>
#include<stdlib.h>
int main( ) {
    int X = 8, Y = 2;

    printf(" A soma de X e Y %c : %d", 130, X + Y);
    printf (" \n A subtracao de X e Y %c : %d",130, X - Y);
    printf (" \nA multiplicacao de X e Y %c : %d", 130, X * Y);
    printf (" \nA divisao de X e Y %c : %.2f", 130, (float) X / Y);
    printf (" \nA divisao de inteiros X e Y %c : %d",130, X / Y);
    printf (" \nO valor de X %c : %d",130, X);
    printf (" \nO incremento do valor de X %c : %d",130, X++);
    printf (" \nO novo valor de X %c : %d",130, X);
    printf (" \nO valor de Y %c : %d",130, Y);
    printf (" \nO incremento do valor de Y %c : %d \n",130, ++Y);

    system("pause");
    return (0);
}
```

Laboratório de Programação

Operadores Aritméticos

- **Módulo**
 - **Operação binária**
 - É avaliado como o resto da divisão integral dos operandos.
 - Com tipos integrais, tem valor integral.
 - Não existe para tipos de ponto flutuante.

Exemplo:

“6 % 2” (avaliado como “0”)

“19 % 3” (avaliado como “1”)

Laboratório de Programação

Operadores Aritméticos

- **Atribuição**
 - **Operação binária**
 - O operando da esquerda é modificado para o valor do operando da direita.

Exemplo:

“num = 2”

“i = 0”

“tentativas = 40”

“a = b + c”

Laboratório de Programação

Operadores Aritméticos

- **Atribuição**

- Podem ser abreviadas as atribuições que tem à direita uma expressão que contém o operando da esquerda.
- A abreviação fica na forma “variavel operador= expressão”

Exemplo:

“A += B” (idêntica a “A = A + B”)

“i += 1” (“i = i + 1”)

“cota /= 2” (“cota = cota / 2”)

Laboratório de Programação

Operadores Relacionais

- Representam Relações binárias sobre números, avaliadas como Verdadeiro ou Falso*

Operador	Relação
>	Maior-que
>=	Maior-ou-igual-a
<	Menor-que
<=	Menor-ou-igual-a
==	Igual-a
!=	Diferente de

*: Em C, Falso é representado pelo valor numérico "0", e Verdadeiro por um outro valor qualquer.

Existem as constantes TRUE e FALSE para esses valores.

Laboratório de Programação

Operadores Relacionais

- **$>$, \geq , $<$ e \leq**
 - Operações binárias
 - **Serão avaliados como Verdadeiro, se os valores concordarem com a expressão. Caso contrário, serão avaliados como Falso.**

Exemplo:

$“3 > 4”$ (Falso)

$“5 \geq -1”$ (Verdade)

$“i < 100”$ (Depende de i)

Laboratório de Programação

Operadores Relacionais

- **== e !=**
 - Operações binárias
 - Avaliam se os operandos são iguais ou diferentes.

Exemplo:

“A == B” (Depende das variáveis)

“i != 0” (Depende de i)

“5 != 7” (Verdade)

“200 == 3” (Falso)

Laboratório de Programação

Operadores Lógicos

- Operam sobre valores Verdadeiro ou Falso, e são avaliados também como Verdadeiro ou Falso.

Operador	Relação
&&	E (AND)
	OU (OR)
!	NÃO (NOT)

- E (AND) será Verdade se os dois operandos forem Verdade.
- OU (OR), se algum dos dois forem Verdade.
- NÃO (NOT), se o operando for Falso.

Exemplo:

“A && B” (Depende de A e B)

“!FALSE” (Verdade)

Laboratório de Programação

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).`

Laboratório de Programação

Operadores

Precedência dos Operadores

Maior	() [] -> .
	! ~ ++ -- (type) * & sizeof
	* / %
	+ -
	<< >>
	< <= > >=
	== !=
	&
	^
	
	&&
	
	?:
Menor	= += -= *= /=

Laboratório de Programação

Operadores de Atribuição

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 $x = x \% y$

Laboratório de Programação

Operadores de Atribuição

Exemplo:

```
#include<stdio.h>
#include<stdlib.h>
int main( ) {
    int X = 8, Y = 2, AUX = 0;

    printf (" \n O valor de X %c : %d",130, X);
    printf (" \n O valor de Y %c : %d",130, Y);
    AUX = X;
    printf (" \n Equivale a X = X + Y: %d", AUX += Y);
    AUX = X;
    printf (" \n Equivale a X = X - Y: %d", AUX -= Y);
    AUX = X;
    printf (" \n Equivale a X = X * Y: %d", AUX *= Y);
    AUX = X;
    printf (" \n Equivale a X = X / Y: %d", AUX /= Y);
    printf (" \n O resto da divisão de X por Y %c : %d ",130, X % Y);
    printf (" \n O resto da divisão de X por Y %c : %d \n",130, X % Y);
    AUX = X;
    printf (" \n Equivale a X = X / Y: %d \n", AUX /= Y);
    system("pause");
    return (0);
}
```


Laboratório de Programação

Operadores com Caracteres Especiais

<code>\n</code>	nova linha
<code>\r</code>	“enter”
<code>\a</code>	bipe
<code>\t</code>	tabulação horizontal (tab)
<code>\v</code>	tabulação vertical
<code>\b</code>	retrocesso
<code>\“</code>	aspas
<code>\\</code>	barra
<code>\?</code>	ponto de interrogação
<code>%%</code>	caractere %

Laboratório de Programação

Operadores com Caracteres Especiais

Exemplo:

```
#include<stdio.h>
#include<stdlib.h>
int main( ) {

    printf (" \n ");
    printf (" \n\n ");
    printf (" \t TESTE \n");
    printf (" \t\t\t TESTE \n");
    printf (" \a \a ");
    printf (" \"TESTE\" \n");
    printf (" \\TESTE\\ \n");
    printf (" \t\t TESTE\? \n");
    printf (" \n TESTE 10%% \n\n");

    system("pause");
    return (0);
}
```

Laboratório de Programação

Funções Matemáticas

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

`#include <math.h>`

Laboratório de Programação

Funções Matemáticas

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main( ) {
    int X = 8, Y = 2;
    double Z = 8.3, W = 5.7;

    printf (" \n O valor de X %c : %d",130, X);
    printf (" \n O valor de Y %c : %d",130, Y);
    printf (" \n O valor de Z %c : %f",130, Z);
    printf (" \n O valor de W %c : %f",130, W);
    printf (" \n Numero Z real para cima %c: %f",130, ceil(Z));
    printf (" \n Numero W real para baixo %c: %f",130, floor(W));
    printf (" \n Numero W real para cima %c: %f",130, ceil(W));
    printf (" \n Numero Z real para baixo %c: %f",130, floor(Z));

    printf (" \n O valor de X elevado a Y %c : %f \n",130, pow(X,Y));
    printf (" \n A raiz quadrada de X * 2: %f \n", sqrt(X*2));
    X = X + 17 + Y;
    printf (" \n A raiz cúbica de X: %f \n", cbrt(X));

    system("pause");
    return (0); }
```

Novos Exercícios – Construir programas:

1 – Faça um algoritmo para calcular a média final dos alunos da 5ª Série. Os alunos realizarão cinco provas: P1, P2, P3, P4 e P5.


2 – Ler dois valores A e B, efetuar as operações de adição, subtração, multiplicação e divisão de A por B, apresentando no final os quatro resultados obtidos.

3 – Escreva um algoritmo que leia dois valores A e B, para calcular: $C = (A + B) * B$. No final apresente o resultado.

4 – Leia 2 variáveis A e B, que correspondem a 2 notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 3,5 e a nota B tem peso 7,5. No final apresente a média do aluno.

5 – Ler dois valores A e B, efetuar a troca dos valores de forma que a variável A passe a possuir o valor da variável B e que a variável B passe a possuir o valor da variável A. Apresentar os valores trocados.

6 – Faça um algoritmo que leia 4 variáveis A,B,C e D. A seguir, calcule e mostre a diferença do produto de A e B pelo produto de C e D. Fórmula da diferença do produto $(A*B)-(C*D)$.



LABORATÓRIO DE PROGRAMAÇÃO

Tipos de Dados e Operadores em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.
Cursos de Computação
1º. Período