

Introdução à Ciência da Computação

Disciplina: 113913

Prof. Luiz Augusto F. Laranjeira

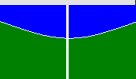
Universidade de Brasília – UnB
Campus Gama



4. Tipos de dados, variáveis e operadores

Prof. Luiz Augusto F. Laranjeira

Universidade de Brasília – UnB
Campus Gama



Tipos de Dados – Dados Numéricos

■ Numéricos inteiros

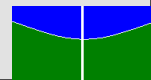
- Os números inteiros são aqueles que não possuem componentes decimais ou fracionários, podendo ser positivos ou negativos.

Exemplos de números inteiros:

36	número inteiro positivo;
0	número inteiro;
-18	número inteiro negativo.

Números inválidos:

12,243	carácter ilegal (,)
36.0	carácter ilegal (.)
10 2	carácter ilegal (espaço)
123-45	carácter ilegal (-)
083	o primeiro algarismo não pode ser zero



Tipos de Dados – Dados Numéricos

- Numéricos reais
 - Os dados de tipo real são aqueles que podem possuir componentes decimais ou fracionários, e podem também ser positivos ou negativos.

Exemplos de números reais:

36.01	número real positivo com duas casas decimais;
166.	número real positivo com zero (nenhuma) casa decimal;
-18.8	número real negativo com uma casa decimal;
0.0	número real com uma casa decimal;
0.	Número real com zero (nenhuma) casa decimal.

Números inválidos:

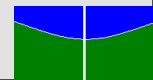
1	tem que ter um ponto
1,03	carácter ilegal (,)
2E 10	carácter ilegal (espaço)
2e5.7	o expoente deve ser uma quantidade inteira

Expressões e Operandos

■ Expressões Aritméticas

- São aquelas cujo resultado da avaliação é do tipo numérico, seja ele inteiro ou real. Somente o uso de operadores aritméticos e variáveis numéricas é permitido em expressões deste tipo.

Operador	Tipo	Operação	Exemplo
-	Unário	Inversão de Sinal	$-(-1) = 1$
+	Unário	Manutenção de Sinal	$+1 = 1$
^	Binário	Exponenciação	$\text{pow}(9,3)$
*	Binário	Multiplicação	$6 * 7 = 42$
/	Binário	Divisão	$8 / 2 = 4$ $8 / 5 = 1$
+	Binário	Adição	$1 + 2 = 3$
-	Binário	Subtração	$4 - 5 = -1$
%	Binário	Resto de Divisão	$8 \% 5 = 3$



Linguagem C – Programa exemplo 1

```
#include <stdio.h> // ?
#include <stdlib.h> // ?
#include <math.h> // ?
// Cabeçalho ? Universidade, aluno, matricula, objetivos do programa
int main()
{
    // Declaração de variáveis
    int x, y, z, soma, sub, mantsin, invsin, mult, pot;
    float modulo, div;
    // Inicialização de variáveis
    x=64; // ?
    y=24; // ?
    z=2; // ?
    // Processamento
    soma=(int) x+y;          sub=(int) x-y;
    pot= (int) pow(z,3);      mantsin=(int) +y;
    invsin=(int) -x;
    mult=(int) x*z;
    div=(float) y/z;
    modulo=(float) (x % y);
    // Apresentação de resultados
    printf ("\n%d + %d = %d",x,y,soma); //Imprime a soma x+y
    printf ("\n%d - %d = %d",x,y,sub); //Imprime a subtração x-y
    printf ("\n%d^3 = %d",z,pot); //Imprime z ao cubo
    printf ("\n+ (%d) = %d",y,mantsin); //Imprime a manutenção do sinal y
    printf ("\n- (%d) = %d",x,invsin); //Imprime a inversão do sinal x
    printf ("\n%d / %d = %.2f",y,z,div); //Imprime a divisão y/z
    printf ("\n%d mod %d = %.2f \n",x,y,modulo); //Imprime o resto da divisão
    printf ("%d\n\n",pot);
    system ("pause");
}
```

Linguagem C – Programa exemplo 1

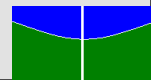
Programa Exemplo em C

Saída do programa:

```
64 + 24 = 88
64 - 24 = 40
2^3 = 8
+ (24) = 24
- (64) = -64
24 / 2 = 12.00
64 mod 24 = 16.00
8Press any key to continue . . .
```

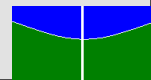
A função printf()

- A função **printf()** serve para mostrar mensagens e os conteúdos de variáveis para o usuário.
- Para mostrar uma mensagem constante:
 - Ex: `printf("Olá mundo!\n");`
- O caractere '\n' faz o programa saltar uma linha ao final da frase.
- Para mostrar o conteúdo de variáveis, é necessária a inserção de **modificadores** na string do printf:
 - Ex: `printf("Idade do funcionário: %d\n",idade);`
- Após a string uma lista das variáveis a serem mostradas deverão corresponder à ordem em que seus modificadores foram inseridos. O tipo da variável deve ser do tipo equivalente a seu modificador.



Alguns modificadores para a função printf()

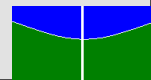
Tipo	Modificador na Linguagem C
int	%d
int com n casas	%nd onde n é o número de casas Ex: %3d para tres casas
int com n casas precedido por zeros	%nd onde n é o número de casas Ex: %03d para tres casas
float	%f
float com n casas decimais	%.nf onde n é o número de casas decimais Ex: %.2f para duas casas decimais
float com m casas ao todo e n casas decimais	%m.nf onde m é o número total de casas e n é o número de casas decimais Ex: %10.2f para dez casas ao todo e duas casas decimais



Exemplos da função printf() com modificadores

```
int    x = 3;  
float  y = 10.1;
```

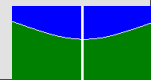
printf() com modificador	Saída
printf("x = %d", x)	x = 3
printf("x = %3d", x)	x = 3
printf("x = %03d", x)	x = 003
printf("y = %.2f", y)	y = 10.1
printf("y = %6.2f", y)	y = 10.1



A função scanf()

- A função **scanf()** lê um valor digitado pelo usuário
- Ele armazena o que o usuário digitou em uma variável no programa.
- Segue a mesma lógica de modificadores que o printf (usando os mesmos modificadores), mas exige que para todos os tipos exceto string seja passado o **ENDEREÇO** da variável, ou seja, colocando um & antes do nome da variável.
- Ex:

```
printf("Digite sua idade: ");  
scanf("%d", &idade);
```



Linguagem C – Programa exemplo 2

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    float vlrconta, taxamulta, vlrmulta, vlrtotal;
    printf("Insira o valor da conta: ");
    scanf("%f",&vlrconta);
    printf("Insira a taxa da multa (em porcentagem): ");
    scanf("%f",&taxamulta);
    vlrmulta=vlrconta*taxamulta/100;
    vlrtotal=vlrconta+vlrmulta;
    printf("Valor da multa: %.2f reais\n",vlrmulta);
    printf("Valor total: %.2f reais\n",vlrtotal);
    system("pause");
}
```



Linguagem C – Programa exemplo 2

Programa Exemplo em C

Saída do programa:

```
Insira o valor da conta: 10
Insira a taxa da multa (em porcentagem): 10
Valor da multa: 1.00 reais
Valor total: 11.00 reais
Press any key to continue . . .
```



Variáveis

■ Variável

- É um local na memória principal, isto é, um **endereço** que armazena um conteúdo.
- Para facilitar a programação, podemos dar um **nome** a esse endereço.
- O conteúdo de uma variável pode ser de um dos vários **tipos** apresentados anteriormente.
- Embora uma variável possa assumir vários valores, pode-se armazenar apenas um de cada vez.

NOME e TIPO de uma variável:

Uma vez definidos,
não podem ser alterados
no decorrer do algoritmo.

CONTEÚDO da variável:

É um objeto de **constante**
modificação no decorrer do
algoritmo.



Variáveis

- Regras (genéricas) para dar nome às variáveis:
 - O **primeiro caractere** deve ser uma letra;
 - Se houver mais de um caracter, só poderemos usar: **letra ou algarismo**;
 - Nomes de variáveis escritas com letras **maiúsculas** serão diferentes de letras **minúsculas**;
 - Nenhuma **palavra reservada** à ferramenta (linguagem C) poderá ser usada como nome de uma variável;
 - Procure dar nomes significativos para a variável (**mnemônicos**), mas não muito longos.

EXEMPLOS:

nota, a, X, w, X25 (nomes de variáveis válidos);
5B, X – Y, E(13) (nomes de variáveis não válidos);



Tipos de Dados – Dados Lógicos

■ Dados Lógicos

- São caracterizados como tipos lógicos, os dados com valores **verdadeiro** e **falso**, sendo que este tipo de dados poderá representar apenas um dos valores.
- É chamado por alguns de tipo **booleano** (matemático George Boole).

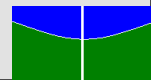


Expressões lógicas

- Operadores relacionais para expressões lógicas:

Operador	Comparação	Linguagem C
=	Igual	==
≠	Diferente	!=

Operadores utilizados para realizar comparações.
Comparações só podem ser feitas entre variáveis do mesmo tipo.
O resultado de uma comparação é sempre um valor lógico.



Linguagem C – Programa exemplo 3

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int x,y;
    x=98;
    y=10;

    printf("Valor 0 indica expressao falsa\n");
    printf("e 1 expressao verdadeira.\n\n");
    printf("%d == %d é %d.\n",x,y,x==y); /* verifica se x é igual a y */
    printf("%d != %d é %d\n",x,y,x!=y); /* verifica se y é diferente de y */
    printf("%d <= %d é %d\n",x,y,x<=y); /* verifica se x é menor ou igual a y */
    printf("%d >= %d é %d\n",x,y,x>=y); /* verifica se x é maior ou igual a y */
    printf("%d < %d é %d\n",x,y,x<y); /* verifica se x é menor que y */
    printf("%d > %d é %d\n",x,y,x>y); /* verifica se y é maior que y */
    system("pause");
}
```



Linguagem C – Programa exemplo 3

Programa Exemplo em C

Saída do programa:

```
Valor 0 indica expressao falsa  
e 1 expressao verdadeira.  
  
98 == 10 eh 0.  
98!=10 eh 1  
98<=10 eh 0  
98>=10 eh 1  
98<10 eh 0  
98>10 eh 1  
Press any key to continue . . . _
```

Tipos de Dados – Dados Literais

■ Dados Literais:

- Uma sequência de caracteres contendo letras, dígitos e/ou símbolos especiais.
- Também chamados de **alfanumérico**, **cadeia de caracteres** ou, ainda, **String**.
- Usualmente são representados nos algoritmos pela coleção de caracteres delimitada em seu início e fim com aspas (“ ”).

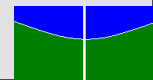
Exemplos de dados literais:

“QUEM?”	Literal de comprimento 5;
“ ”	Literal de comprimento 1;
“cOmO!?#”	Literal de comprimento 7;
“AbcDEFghi”	Literal de comprimento 9;
“4+5-1=”	Literal de comprimento 6;
“1”	Literal de comprimento 1;
“ANA MARIA”	Literal de comprimento 9;

É comum em algumas linguagens a diferenciação entre um caracter (‘**A**’) e um conjunto de caracteres (“**Olá Mundo**”).

Alguns modificadores para a função *printf*

Tipo	Modificador na Linguagem C
int	%d
float	%f (6 decimais) ou %g (até 6 decimais)
double	%lf (6 decimais) ou %lg (até 6 decimais)
float com n casas decimais	%.nf onde n é o número de casas decimais Ex: %.2f para duas casas decimais
double com n casas decimais	%.nlf onde n é o número de casas decimais Ex: %.3lf para tres casas decimais
Literal caracter	char
char	%c
string	%s



Variáveis

■ Declaração de Variáveis

- São declaradas no início (alocação de memória)
- Podem ser dos tipos: inteiro, real, literal, ou lógico
- Permite que o algoritmo saiba os tipos de operações que cada variável pode realizar

Exemplo 1:

Dada uma lista de compras com o código, quantidade, e preço de oito produtos, crie um algoritmo que escreva o valor total da compra.

codigo : literal (ou string)

quantidade : inteiro

preço, valortotal : real



Variáveis

■ Declaração de Variáveis

Exemplo 2:

Ao serem fornecidos um valor a ser pago e uma taxa para multa (o pagamento está sendo feito após o vencimento), calcule o valor da multa e o valor total a ser pago.

vlrconta, taxamulta, vlrmulta, vlrtotal : real

Exemplo 3:

É solicitada a um motorista recém chegado de uma viagem a quantidade de quilômetros por ele percorrida. O motorista informa o solicitado e você deverá informar a ele a quantos metros correspondem à quantidade fornecida em quilômetros.

quilômetros : real

metros : inteiro



Linguagem C – Tipos de Variáveis

Tipo	Linguagem C
Inteiro	int
Real	float (6 decimais de precisão)
Real	double (15 decimais de precisão)
Literal character	char
Literal string	char nome[tamanho] ex: char endereco[30]
lógico	Definir como int, que será “falso” quando for zero, e “verdadeiro” quando não for zero.



Linguagem C – Programa exemplo 4

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int inteiro=4;
    float real=33.67;
    char caractere='j';
    char literal[10]="Jose\0";
    int booleano=1;
    printf("Inteiro: %d\n",inteiro);
    printf("Real: %f\n",real);
    printf("Caractere: %c\n",caractere);
    printf("String: %s\n",literal);
    printf("Verdadeiro\n");
    system("pause");
}
```



Linguagem C – Programa exemplo 4

Programa Exemplo em C

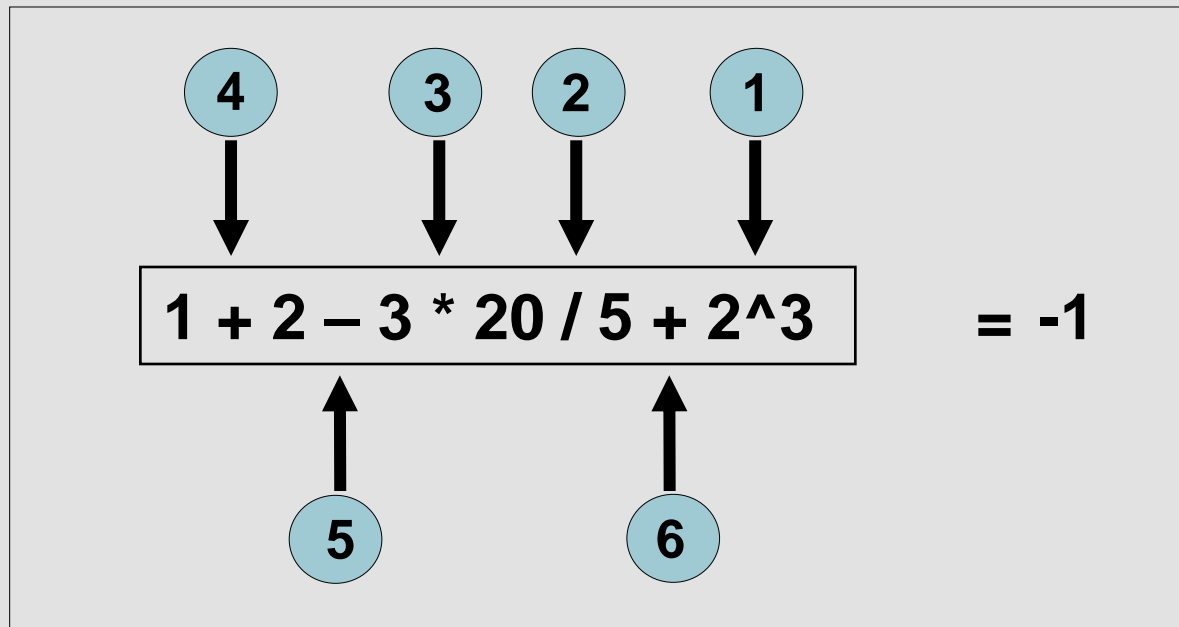
Saída do programa:

```
Inteiro: 4
Real: 33.669998
Caractere: j
String: Jose
Verdadeiro
Press any key to continue . . . _
```

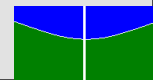


Expressões e Operandos

■ Expressões Aritméticas

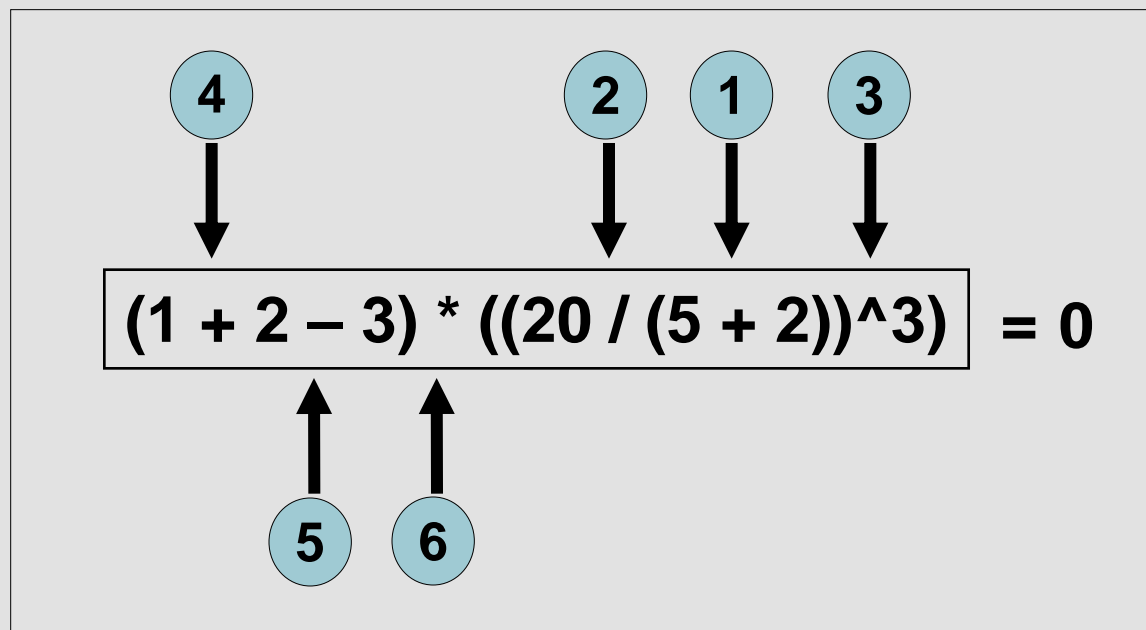


$$(1 + 2) - (3 * (20 / 5)) + (2^3) = -1$$



Expressões e Operandos

- Expressões Aritméticas: Alterando a prioridade das expressões com o uso de PARÊNTESES.



Expressões e Operandos

■ Expressões Lógicas

- São aquelas cujo resultado da avaliação é um valor lógico (VERDADEIRO ou FALSO)

Tabela verdade dos operadores lógicos

		CONJUNÇÃO	DISJUNÇÃO	NEGAÇÃO
x	y	x E y	x OU y	NÃO(x)
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

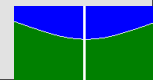


Expressões e Operandos

■ Expressões Lógicas

- Os operadores lógicos também são chamados de operadores booleanos.

Operador	Tipo	Operação	Prioridade	Linguagem C
OU	Binário	Disjunção	3	
E	Binário	Conjunção	2	&&
NÃO	Unário	Negação	1	!



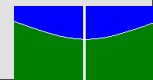
Expressões e Operandos

■ Expressões Lógicas

- **EXEMPLO:** Suponha duas perguntas feitas a seis pessoas que se candidataram a uma entrevista de emprego de programador. As respostas às perguntas serão SIM ou NÃO. Suponha também que só será chamado para a entrevista o candidato que responder SIM às duas perguntas.

Exemplo e tabela verdade do operador E

Candidato	Você conhece a linguagem C?	Você conhece a linguagem Java?	Candidato aprovado para a entrevista?
Moisés	Não	Sim	Não
Rudimar	Não	Sim	Não
Nilton	Sim	Não	Não
Itamar	Sim	Não	Não
Rogério	Sim	Sim	Sim
Zilon	Sim	Não	Não



Expressões e Operandos

■ Expressões Lógicas

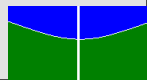
- Operadores relacionais para expressões lógicas:

Operador	Comparação	Linguagem C
=	Igual	==
≠	Diferente	!=
<	Menor	<
<=	Menor ou Igual	<=
>	Maior	>
>=	Maior ou Igual	>=

Operadores utilizados para realizar comparações.

Comparações só podem ser feitas entre variáveis do mesmo tipo.

O resultado de uma comparação é sempre um valor lógico.

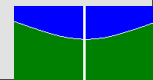


Expressões e Operandos

■ Expressões Lógicas

Exemplo e tabela verdade de operadores relacionais

Exemplo	Valores	Questionamento	Resultado
$A \neq B$	$A = 5$ e $B = 6$	A é diferente de B?	VERDADEIRO
$A \neq B$	$A = 5$ e $B = 5$	A é diferente de B?	FALSO
$X == 1$	$X = 2$	X é igual a 1?	FALSO
$X == 1$	$X = 1$	X é igual a 1?	VERDADEIRO
$7 > 6$		7 é maior que 6?	VERDADEIRO
$8 < 9$		8 é menor que 9?	VERDADEIRO
$1 \leq Y$	$Y = 1$	1 é menor ou igual a Y?	VERDADEIRO
$1 \leq Y$	$Y = 2$	1 é menor ou igual a Y?	VERDADEIRO
$1 \leq Y$	$Y = 0$	1 é menor ou igual a Y?	FALSO
$4 \geq W$	$W = 4$	4 é maior ou igual a W?	VERDADEIRO
$4 \geq W$	$W = 3$	4 é maior ou igual a W?	VERDADEIRO
$4 \geq W$	$W = 5$	4 é maior ou igual a W?	FALSO



Linguagem C - Operadores

Operador	Operação	Linguagem C
=	Igual/Igualdade	==
≠	Diferente/Desigualdade	!=
<	Menor	<
<=	Menor ou Igual	<=
>	Maior	>
>=	Maior ou Igual	>=
OU	Disjunção	
E	Conjunção	&&
NÃO	Negação	!
*	Multiplicação	*
/	Divisão	/
+	Adição	+
-	Subtração	-
mod	Resto da Divisão	%
←	Atribuição	=

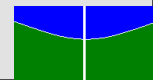


Algumas Funções

Função	Sintaxe	Exemplo	=	Objetivo	Linguagem C
QUOC	QUOC (A,B)	QUOC (10,3)	3	Retornar a parte inteira de uma divisão	Basta a variável que recebe o resultado da operação ser do tipo inteiro.
RESTO()	RESTO(A,B)	RESTO(10,3)	1	Retornar o resto de uma divisão.	%
^	A^B	2^3	8	Elevar um número a um expoente	pow()
FRAC()	FRAC(A)	FRAC(10,345)	0,345	Retornar a parte fracionária de um número real.	Feito através da subtração do valor pela sua parte inteira e armazenando o resultado em um variável real.
INTEIRO()	INTEIRO(A)	INTEIRO(10,345)	10	Retornar a parte inteira de um número real.	Dado um valor real, armazene-o em uma variável inteira e então, apenas a parte inteira deste valor é armazenada.

Algumas Funções

Função	Sintaxe	Exemplo	=	Objetivo	Linguagem C
ARRED()	ARRED(A)	ARRED(10,4)	10	Arredondar um número. Até 0,499999 arredonda para baixo, acima de 0,499999 arredonda para cima.	round() Para arredondamento sempre acima ceil() Para arredondamento sempre abaixo floor()
RAIZ()	RAIZ(A)	RAIZ(16)	4	Retorna a raiz quadrada de um número.	sqrt()



Linguagem C – Programa exemplo 5

Programa Exemplo em C

Operadores de incremento e decremento

O operador de incremento (++) soma 1 ao seu operando enquanto que o de decremento (--) subtrai um. Eles podem ser pré-fixados ou pós-fixados conforme mostra a tabela abaixo:

Operador	Instrução	Pós-fixado	Pré-fixado
++	var = var+1;	var = var++; ou var++;	var = ++var; ou ++var;
--	var = var-1;	var = var--; ou var--;	var = --var; ou --var;

Linguagem C – Programa exemplo

```
#include<stdio.h>

void main()
{
    int var,var2; //declaração de variáveis
    var=5;
    var2=++var;

    printf("var= %d\n", var);
    printf("var2= %d\n", var2);
}
```

```
#include<stdio.h>

void main()
{
    int var,var2; //declaração de variáveis
    var=5;
    var2=var++;

    printf("var= %d\n", var);
    printf("var2= %d\n", var2);
}
```

Linguagem C – Programa exemplo

Programa Exemplo em C

Saída do programa 1:

var=6

var2=6

Saída do programa 2:

var=6

var2=5



A função sizeof()

```
#include<stdio.h>

void main()
{
    float x;
    double y;
    char z;

    //impressão do tamanho de cada variável utilizando o sizeof:
    printf("A variavel x do tipo float tem o tamho=%d bytes\n",sizeof(x));
    printf("A variavel y do tipo double tem o tamho=%d bytes\n",sizeof(y));
    printf("A variavel z do tipo char tem o tamho=%d bytes\n",sizeof(z))
}
```

Linguagem C – sizeof() saída do programa:

A variavel x do tipo float tem o tamanho=4 bytes
A variavel y do tipo double tem o tamanho=8 bytes
A variavel z do tipo char tem o tamanho=1 bytes

