Conversão de Tipos

- **Implícita** → dos tipos menores para os tipos maiores

Exemplos: char → int

int → long int

float → double

Conversão de Tipos

```
#include <stdio.h>
main ()
{
    int a;
    char b;
    b='B';
    printf ("nº de bytes da variavel b = %d\n", sizeof (b));
    printf ("conteudo da variavel b = %d\n", b);
    a=b;
    printf ("nº de bytes da variavel a = %d\n", sizeof (a));
    printf ("conteudo da variavel a = %d\n", a);
}
```

Conversão de Tipos

- **Explícita** → de qualquer tipo para qualquer outros tipos maiores, utilizando cast **(modeladores)**.

Exemplos: int → char

int → short int

double → float

Conversão de Tipos

```
#include <stdio.h>
main ()
{
    int a;
    char b;
    a=120; // mudar esta linha para: a=359;
    printf ("nº de bytes da variavel a = %d\n", sizeof (a));
    printf ("conteudo da variavel a = %d\n", a);
    b=(char)a;
    printf ("nº de bytes da variavel b = %d\n", sizeof (b));
    printf ("conteudo da variavel b = %d\n", b);
}
```

Operadores

A linguagem C disponibiliza o mesmo conjunto de operadores aritméticos, lógicos e relacionais apresentados anteriormente (com a mesma ordem de precedência).

Além do conjunto de operadores vistos, C disponibiliza um conjunto de operadores lógicos bit a bit, possibilitando uma manipulação em mais baixo nível dos dados contidos em variáveis discretas.

UNIVASF

Operadores Lógicos bit a bit

Operador	Ação
&	AND(e)
|	OR(ou)
^	XOR(ou exclusivo)
>>	Desloca os bits “x” vezes a direita
<<	Desloca os bits “x” vezes a esquerda
~	NOT(não)

Obs.: Aplicados a char e int.

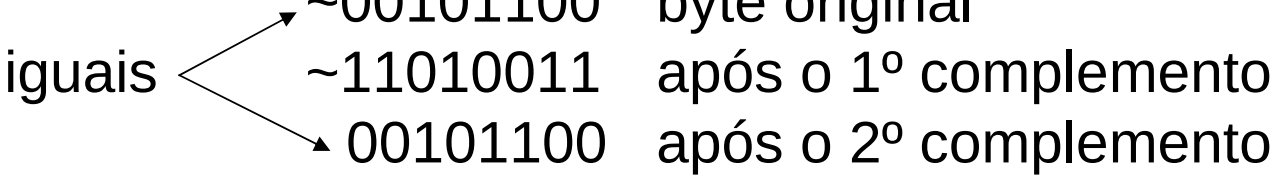
Operadores Lógicos bit a bit

Exemplos:

```
  11000001   193 em binário
  01111111   127 em binário
& ________   AND bit a bit
  01000001   65 em binário

  10000000   128 em binário
  00000011   3 em binário
| ________   OR bit a bit
  10000011   131 em binário

  01111111   127 em binário
  01111000   120 em binário
^ ________   XOR bit a bit
  00000111   7 em binário
```



Operadores Lógicos bit a bit

Exemplos:

- Deslocamento à esquerda:
  variavel2 = variavel1 << num_de_deslocamentos

- Deslocamento à direita
  variavel2 = variavel1 >> num_de_deslocamentos

unsigned char x;	x a cada execução da sentença	Valor de x
x=7;	00000111	7
x=x<<1;	00001110	14
x=x<<3;	01110000	112
x=x<<2;	11000000	192
x=x>>1;	01100000	96
x=x>>2;	00011000	24

Funções de Entrada e Saída Formatada

#include <stdio.h>

- std → standard
- io → input/output

printf ()

- Forma geral:

  printf (string_de_controle<,lista_de_argumentos>);

Funções de Entrada e Saída Formatada

- printf (continuação)
  - *string_de_controle*
    - descrição de tudo que a função colocará na tela;
    - indica os caracteres;
    - indica as variáveis com suas respectivas posições. Isso é feito usando-se os códigos de controle, que usam a notação do %.

Funções de Entrada e Saída Formatada

Tabela de códigos de formato (%)

Código	Formato
%c	Um caracter (char)
%d	Um número inteiro decimal (int)
%i	O mesmo que %d
%e	Número em notação científica com o "e"minúsculo
%E	Número em notação científica com o "e"maiúsculo
%f	Ponto flutuante decimal
%g	Escolhe automaticamente o melhor entre %f e %e
%G	Escolhe automaticamente o melhor entre %f e %E
%o	Número octal
%s	String
%u	Decimal "unsigned" (sem sinal)
%x	Hexadecimal com letras minúsculas
%X	Hexadecimal com letras maiúsculas
%%	Imprime um %
%p	Ponteiro

Funções de Entrada e Saída Formatada

- printf (continuação)
  - *lista_de_argumentos*

    Para cada código % contido na string de controle, temos um argumento correspondente na *lista_de_argumentos*

Funções de Entrada e Saída Formatada

- printf (continuação)

Vamos ver alguns exemplos:

Código	Imprime
printf ("Um %%%c indica %s",'c',"char");	Um %c indica char
printf ("%X %f %e",107,49.67,49.67);	6B 49.670000 4.967000e+001
printf ("%d %o",10,10);	10 12

Exercício

Construa um programa que escreva a string “juros de” e o inteiro 10 na tela, constituindo a seguinte frase:

Juros de 10%

Funções de Entrada e Saída Formatada

- printf (continuação)

É possível também indicar o tamanho do campo, justificação e o número de casas decimais. Para isso utiliza-se códigos colocados entre o % e a letra que indica o tipo do formato.

Exemplos:

%5d, %05d, %-5d

%10.4f, %-10.15s, %.4g

Funções de Entrada e Saída Formatada

- printf (continuação)

Exercício: Construa um programa em C que utilizando-se dos códigos % escreva na saída padrão a seguinte seqüência no formato apresentado: