Operadores

- Operadores Aritméticos
  - Unários: +, -, ++, --

Exemplos:

+1

-5

a=-b;

a++; ⇔ a=a+1;

a--; ⇔ a=a-1;

b=a++; ⇔ b=a;
a=a+1;

b=++a; ⇔ a=a+1;
b=a;

Obs.: a e b são variáveis numéricas.

UNIVASF

Operadores

- Operadores Aritméticos
  - Binários: +, -, *, /, %

10/3=? 3
10/3.0=? 3.3333
10.0/3=? 3.3333
10.0/3.0=? 3.3333

Não existe o operador \ (trabalhado em Introdução a Algoritmos)

Precedência (Hierarquia nas operações)

Hierarquia	Operação
1	Parênteses
2	Função
3	++,--
4	- (menos unário)
5	*, /, %
6	+, -

Operadores

- Operadores de Atribuição

  =, +=, -=, *=, /=, %=

  Exemplos:

  a=5;

  a+=5; ⇔ a=a+5;

  a*=5-2; ⇔ a=a*(5-2);

Operadores

Operadores Lógicos

Operador	Ação
&&	e
||	ou
!	não

&&	V	F
V	V	F
F	F	F

||	V	F
V	V	V
F	V	F

!V == F

Operadores

Operadores Relacionais

Operador	Ação
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual
==	igual a
!=	diferente de

Operadores

Precedência (Hierarquia dos operadores relacionais e lógicos)

Hierarquia	Operação
1	!
2	>, >=, <, <=
3	==, !=
4	&&
5	||

Operadores Lógicos bit a bit

Obs.: Aplicados a char e int.

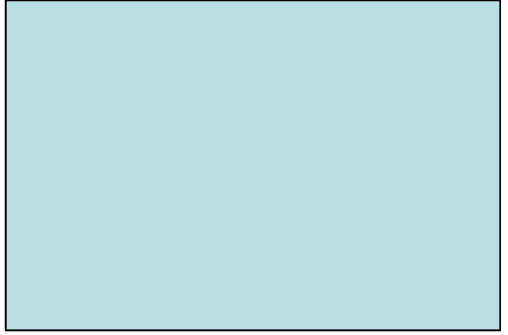
Operador	Ação
&	AND(e)
|	OR(ou)
^	XOR(ou exclusivo)
~	NOT(não)
>>	Desloca os bits à direita
<<	Desloca os bits à esquerda

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

  01111110   126 em binário
  01111001   121 em binário
^ ________   XOR bit a bit
  00000111   7 em binário
```



$1\ 1\ 0\ 0\ 0\ 0\ 0\ 1 =$

$2^7*1+2^6*1+2^5*0+2^4*0+2^3*0+2^2*0+2^1*0+2^0*1 =$

$128 + 64 + 0 + 0 + 0 + 0 + 0 + 1 = 193$

5 && 2 == 5 & 2

?

~00101100 byte original (44 em binário)
~11010011 após o 1º complemento
 00101100 após o 2º complemento

UNIVASF

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 na base binária a cada execução da sentença	Valor de x na base decimal a cada execução da sentença
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

UNIVASF

Primeiro Programa na Linguagem C
Funções de Entrada e Saída Formatada

Primeiro programa na linguagem C

- **Estrutura mínima**

```
main()
{
   ...

}
```

Primeiro programa na linguagem C

- **Estrutura mínima**

```
main()
{
  int x = -1, y;
  y = x + 7;
}
```

Funções de Entrada e Saída Formatada

Conceito de função:

Ação, ato ou efeito.

Exemplo:

Dentre as funções dos vigias da UNIVASF, está a ***função*** de abrir a porta das salas de aula.

Contudo, para que um dos vigias abra a porta de uma das salas de aula temos que solicitar ao mesmo e informar, durante a solicitação, a data, o horário e o número da sala a ser aberta.

Computacionalmente, em uma análise inicial, uma função representa uma sequência de instruções que será executada para efetuar uma certa tarefa. Denominamos como ***parâmetros*** da função os dados necessários para que esta execute.

UNIVASF

Funções de Entrada e Saída Formatada

Diretiva do pré-processador

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

- std → standard (padrão)
- io → input/output (entrada/saída)

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 apresentará na tela;
    - indica os caracteres;
    - indica as variáveis e expressões, cujos valores serão apresentados, e em quais posições específicas aparecerão. Isso é feito, usando os códigos de controle, com a notação %.

UNIVASF

Funções de Entrada e Saída Formatada

Tabela de códigos de formato (%)

Código	Formato
%c	Um caractere (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/menor entre %f e %e
%G	Escolhe automaticamente o melhor/menor 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 %

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

Funções de Entrada e Saída Formatada

- printf (continuação)

/* Estrutura geral de um programa em C que utiliza a função de saída formatada */

```
#include <stdio.h>
main ()
{
        ...
        printf("...",...);
        ...
}
```

Funções de Entrada e Saída Formatada

- printf (continuação)

/* Estrutura geral de um programa em C que utiliza a função de saída formatada */

```
#include <stdio.h>
main ()
{
        char c;
        c = 'A';
        printf("%c",c);
}
```

Funções de Entrada e Saída Formatada

Exercício

Construa um programa, na linguagem de programação C, que escreva a string “juros de ”, o inteiro 10 e o caractere ‘%’ na tela, constituindo a seguinte frase:

juros de 10%