Operadores

Operadores

- Os operadores são usados para desenvolver operações matemáticas.
- Podem ser
  - Unários
  - Binários
  - Bit a bit

Operadores

- Operadores Unários

+ : mais unário ou positivo	+x
- : menos unário ou negação	-x
! : NOT ou negação lógica	!x
&: endereço	&x
*: conteúdo (ponteiros)	(*x)
++: pré ou pós incremento	++x ou x++
-- : pré ou pós decremento	-- x ou x --

Operadores

- Diferença entre pré e pós incremento/decremento
  - y = x++: incrementa depois de atribuir
  - y = ++x: incrementa antes de atribuir

Operadores

- Ex:

```
int x,y;
x = 10;
y = x++;
printf("%d \n",x);
printf("%d \n",y);
y = ++x;
printf("%d \n",x);
printf("%d \n",y);
```

- Resultado

```
11
10
12
12
```

Operadores

- Binários
  - +: adição de dois números z = x + y;
  - - : subtração de dois números z = x – y;
  - * : multiplicação de dois números z = x * y;
  - / : quociente de dois números z = x / y;
  - %: resto da divisão: z = x % y;

Expressões

- Expressões são combinações de variáveis, constantes e operadores.
- Exemplos:

```
Anos = Dias/365.25;
i = i+3;
c= a*b + d/e;
c= a*(b+d)/e;
```

Modeladores (Casts)

- Um modelador é aplicado a uma expressão.
- Força o resultado da expressão a ser de um tipo especificado.
  - (tipo) expressão
- Ex:
  - (float) x;
  - (int) x * 5.25;

Modeladores (Casts)

- Ex:

```
int num;
float f;
num = 10;
f = num/7;
printf ("%f \n", f);
f = (float)num/7;
printf ("%f", f);
```

- Resultado

1.000000
1.428571

Operadores Simplificados

- O C permite simplificar algumas expressões matemáticas

+= : soma e atribui	x += y;	x = x + y;
-= : subtrai e atribui	x -= y;	x = x - y;
*= : multiplica e atribui	x *= y;	x = x * y;
/= : divide e atribui quociente	x /= y;	x = x / y;
%= : divide e atribui resto	x %= y;	x = x % y;
&= : E bit-a-bit e atribui	x &= y;	x = x & y;
|= : OU bit-a-bit e atribui	x |= y;	x = x | y;
<<= : shift left e atribui	x <<= y;	x = x<<y;

Operadores

- As operações bit-a-bit ajudam programadores que queiram trabalhar com o computador em "baixo nível".
- Essas operações só podem ser usadas nos tipos char, int e long.

Operadores

- Operações bit-a-bit: o número é representado por sua forma binária e as operações são feitas em cada bit dele.
  - <<: desloca à esquerda x << 2
  - >>: desloca à direita x >>2
  - ^ : ou exclusivo x ^ 0xF0
  - & : E bit-a-bit x & 0x07
  - | : OU bit-a-bit x | 0x80
  - ~ : Complemento bit-a-bit ~ x

Operadores

- Operador **E** bit a bit
  - char x = 0xD5;

1	1	0	1	0	1	0	1

  - 0x0F

0	0	0	0	1	1	1	1

  - x & 0x0F

0	0	0	0	0	1	0	1

Operadores

- Operador **OU** bit a bit
  - char x = 0xD5;

1	1	0	1	0	1	0	1

  - 0x0F

0	0	0	0	1	1	1	1

  - x | 0x0F

1	1	0	1	1	1	1	1

Operadores

- Operador **OU EXCLUSIVO** bit a bit
  - char x = 0xD5;

1	1	0	1	0	1	0	1

  - 0x0F

0	0	0	0	1	1	1	1

  - x ^ 0x0F

1	1	0	1	1	0	1	0

Operadores

- Complemento bit a bit
  - char x = 0xD5;

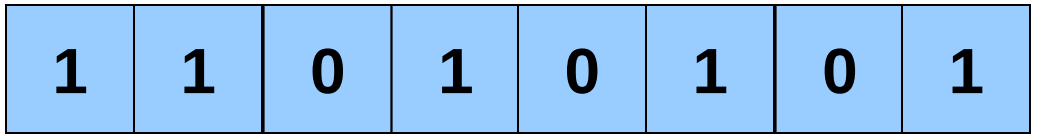
1	1	0	1	0	1	0	1

  - ~x

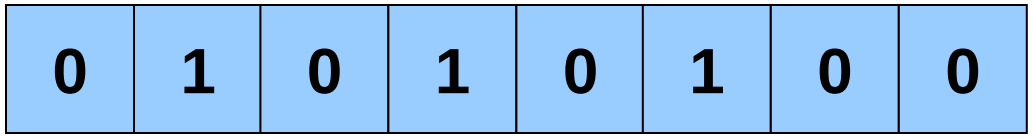
0	0	1	0	1	0	1	0

Operadores

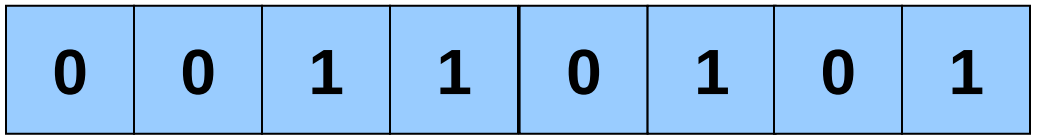
- Deslocamento bit a bit
  - char x = 0xD5;



  - x << 2
    - (x*4)



  - x >> 2
    - (x/4)



Overflow e underflow

- Quando representamos um valor menor ou maior que o permitido pelo tipo, ocorre um erro de cálculo
  - Overflow: valor superior ao permitido
  - Underflow: valor inferior ao permitido

Exemplo: Underflow

```
#include<stdio.h>

int main() {
  short valor = -32768;
  printf("Valor antes: %d\n",valor);//Imprime -32768
  valor--;//Gerando um underflow
  printf("Valor depois: %d\n",valor);//Imprime 32767
  return 0;
}
```

Exemplo: Overflow

```
#include<stdio.h>

int main() {
  short int valor = 30000;
  printf("Valor antes: %d\n",valor);//Imprime 30000
  valor *= 10;//Gerando um overflow
  printf("Valor depois: %d\n",valor);//Imprime -27680
  return 0;
}
```

Overflow e underflow

- Não dão erro: o programa **continua a execução** na **maioria das linguagens de programação**

- **Moral da história:**
  - Procure saber quais são os valores máximos e mínimos que seu programa deve suportar, e escolha o tamanho da variável de acordo

Operadores Relacionais e Lógicos

- Operadores relacionais: comparação entre variáveis

>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

- Esse tipo de operador retorna *verdadeiro* (1) ou *falso* (0)

Operadores Relacionais e Lógicos

- Exemplos
  - Expressão — Resultado
  - x=5; x > 4 — verdadeiro (1)
  - x=5; x == 4 — falso (0)
  - x=5; y=3; x != y — verdadeiro (1)
  - x=5; y=3; x != (y+2) — falso (0)

Operadores Relacionais e Lógicos

- Operadores lógicos: retornam um valor lógico *verdadeiro* (1) ou *falso* (0)

Operador	**Função**	**Exemplo**
&&	AND (E)	(c >= ‘0’ && c<= ‘9’)
||	OR (OU)	(a== ‘F’ || b!=32)
!	NOT	(NÃO)

Operadores Relacionais e Lógicos

- Tabela verdade

a	b	!a	!b	a && b	a || b
0	0	1	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	1