

Câmpus Charqueadas

Operadores Aritméticos e Lógicos em C

Programação Estruturada

Prof. André del Mestre

Operadores para acumulação

- Destaque para os novos operadores!
 - Exclusivos para ACUMULACAO!

Operação	Tradicionais	Exemplo	Novos	Exemplo
Soma	+	a=a+X;	+=	a += X;
Subtração	-	b=b-Y;	-=	b-=Y;
Multiplicação	*	c=c*4;	*=	c *=4 ;
Divisão	/	d=d/3;	/=	d /=3 ;
Resto de divisão	%	e=e%2;	%=	e%=2;



Operadores para acumulação

Exemplo 1

```
ac = 7
ac = 3
ac = 1
```

```
#include <stdio.h>
int main (){
   int ac=1;
   ac *=7;
   printf("ac = %i\n", ac);
   ac/=2;
   printf("ac = %i\n", ac);
   ac%=2;
   printf("ac = %i\n", ac);
   return 0;
```



Operadores para acumulação

- Destaque para os novos operadores!
 - Exclusivos para ACUMULACAO!

Operação	Pós-fixado	Pré-fixado	Equivalênc	ias
Incrementa	i++;	++i;	i=i+1;	i +=1 ;
Decrementa	i;	i;	i=i-1;	i-=1;



Operadores para acumulação

• Exemplo 2

```
cont=1
cont=2
cont=3
```

```
#include <stdio.h>
int main (){
   int cont=0;
   cont=cont+1;
   printf("cont= %i\n", cont);
   cont+=1;
   printf("cont= %i\n", cont);
   cont++;
   printf("cont= %i\n", cont);
   return 0;
```



Operadores para acumulação

• Exemplo 3

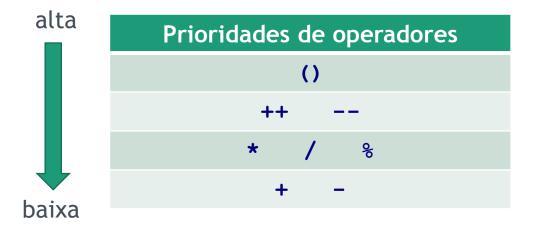
```
    i = 9
    i = 9
    i = 8
    i = 7
```

```
#include <stdio.h>
int main (){
   int i=10;
   i--;
   printf("i= %i\n", i);
   printf("i= %i\n", i--);
   printf("i= %i\n", i);
   printf("i= %i\n", --i);
   return 0;
```



Precedência de operadores

Quanto eh dois mais dois dividido por dois?



$$2 + 2 / 2 = 3$$

 $(2 + 2) / 2 = 2$

Entendendo V ou F

- Linguagem C não tem o tipo booleano!
 - FALSO -> 0
 - VERDADEIRO -> qualquer coisa diferente de 0
 - Operadores booleanos retornam 1 para VERDADEIRO

Operação	Operador
Eh igual?	==
Eh diferente?	!=
Eh maior/menor que?	> <
Eh maior/menor OU igual que?	>= <=



Entendendo V ou F

Exemplo

```
V ou F? 1
V ou F? 0
V ou F? 0
V ou F? 0
```

```
#include <stdio.h>
int main (){
   int a=2, b=3, c;
   c=0!=1;
   printf("V ou F? %i\n", c);
   c=a>2;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", 1>=2);
   printf("V ou F? %i\n", a==b);
                                 EDUCAÇÃO
   return 0;
```

Entendendo V ou F

- Linguagem C não tem o tipo booleano!
 - FALSO -> 0
 - VERDADEIRO -> qualquer coisa diferente de 0
 - Operadores booleanos retornam 1 para VERDADEIRO

Operação	Operador
Negação	!
E lógico	& &
OU lógico	11

```
#include <stdio.h>
int main () {
   int c=5;

RO
   c = !c;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", -1 && c);
   printf("V ou F? %i\n", c || -4);

   return 0;
}
```



Entendendo V ou F

NEGACAO - tabela verdade

Entrada	Saida
V	F
F	V

```
#include <stdio.h>
int main () {
   int c=5;

   c = !c;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", -1 && c);
   printf("V ou F? %i\n", c || -4);

   return 0;
}
```



Entendendo V ou F

• E - tabela verdade

Entradas		Saida
V	V	V
V	F	F
F	V	F
F	F	F

```
#include <stdio.h>
int main () {
   int c=5;

   c = !c;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", -1 && c);
   printf("V ou F? %i\n", c || -4);

   return 0;
}
```



Entendendo V ou F

• OU - tabela verdade

Entr	Entradas	
V	V	V
V	F	V
F	V	V
F	F	F

```
#include <stdio.h>
int main () {
   int c=5;

   c = !c;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", -1 && c);
   printf("V ou F? %i\n", c || -4);

   return 0;
}
```



Entendendo V ou F

```
V ou F? 0
V ou F? 0
V ou F? 1
```

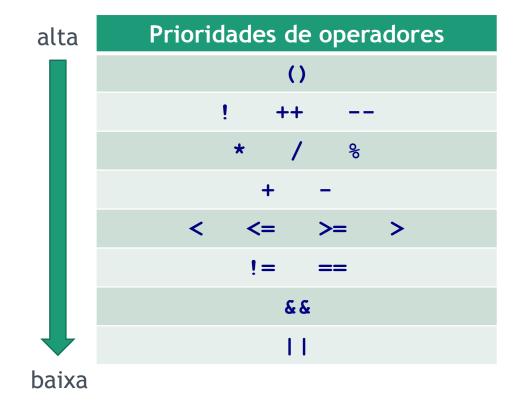
```
#include <stdio.h>
int main () {
   int c=5;

   c = !c;
   printf("V ou F? %i\n", c);
   printf("V ou F? %i\n", -1 && c);
   printf("V ou F? %i\n", c || -4);

   return 0;
}
```



Precedência de Operadores





MUITO OBRIGADO

Prof. André del Mestre

www.ifsul.edu.br almmartins@charqueadas.ifsul.edu.br