



Modificadores de Tipos

Modificadores de Tipos

Palavras reservadas:

- **signed (com sinal)**
- **unsigned (sem sinal)**
- **long (expande faixa de valores)**
- **short (contraí faixa de valores)**

Exemplo: unsigned char letra;

long int numero1, numero2;

Tipos de dados definidos no padrão ANSI

Tipo	Tamanho aproximado em bits	Faixa mínima
char	8	-127 a 127
unsigned char	8	0 a 255
signed char	8	-127 a 127
int	16	-32.767 a 32.767
unsigned int	16	0 a 65.535
signed int	16	O mesmo que int
short int	16	O mesmo que int
unsigned short int	16	0 a 65.535
signed short int	16	O mesmo que short int
long int	32	-2.147.483.647 a 2.147.483.647
signed long int	32	O mesmo que long int
unsigned long int	32	0 a 4.294.967.295
float	32	Seis dígitos de precisão
double	64	Dez dígitos de precisão
long double	80	Dez dígitos de precisão

Fonte: livro "C Completo e Total"

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

```
int main()
```

```
{
```

```
    char c;
```

```
    int i;
```

```
    short int si;
```

```
    unsigned int ui;
```

```
    long int li;
```

```
    float f;
```

```
    double d;
```

```
    printf("char %d \n", sizeof(c));
```

```
    printf("int %d \n", sizeof(i));
```

```
    printf("short int %d \n", sizeof(si));
```

```
    printf("unsigned int %d \n", sizeof(ui));
```

```
    printf("long int %d \n", sizeof(li));
```

```
    printf("float %d \n", sizeof(f));
```

```
    printf("double %d \n", sizeof(d));
```

```
    printf("double %d", sizeof(double));
```

```
}
```

Função que recebe a palavra reservada referente a um tipo primitivo (com ou sem modificador de tipo) ou um identificador de uma variável e retorna o número de bytes reservado na memória para o armazenamento da mesma.



Conversão de Tipos

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

Exemplos:

char → int

int → long int

float → double

```
#include <stdio.h>
int 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 tipos maiores para tipos menores, utilizando *cast* (**modelador**).

Exemplos:

int → char

double → float

```
#include <stdio.h>
int main ()
{
    int a;
    char b;
    a=120; /*120(DEC) == ...0000000001111000(BIN)*/
    printf ("nº de bytes da variavel a = %ld\n", sizeof (a));
    printf ("conteudo da variavel a = %d\n", a);
    b=(char)a; /*Sem perda de dados. Pois, 01111000(BIN) == 120(DEC)*/
    printf ("nº de bytes da variavel b = %ld\n", sizeof (b));
    printf ("conteudo da variavel b = %d\n", b);
}
```



```
#include <stdio.h>
int main ()
{
    int a;
    char b;
    a=365; /*365(DEC) == 101101101(BIN)*/
    printf ("nº de bytes da variavel a = %ld\n", sizeof (a));
    printf ("conteudo da variavel a = %d\n", a);
    b=(char)a; /*Com perda de dados. Pois, 01101101(BIN) == 109(DEC)*/
    printf ("nº de bytes da variavel b = %ld\n", sizeof (b));
    printf ("conteudo da variavel b = %d\n", b);
}
```

Exercício

Construa um programa, na linguagem C, que receba através da entrada padrão um valor inteiro pertencente ao intervalo $[0,255]$ e a posição do bit que se deseja saber o valor, o programa deve escrever na saída padrão o valor do bit solicitado.

Exercício

[0,255]

$$137 = \begin{array}{ccccccc} \underline{1} & \underline{0} & \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{array}$$

&

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} >>$$

3 

```
#include <stdio.h>
int main ()
{
    int desloca, valor_byte, aux=1;
    printf("\nDigite um numero pertencente ao intervalo [0,255]: ");
    scanf("%d", &valor_byte);
    printf("\nDigite o bit a testar(0 a 7): ");
    scanf("%d", &desloca);
    aux = aux << desloca;
    valor_byte = valor_byte & aux;
    valor_byte = valor_byte >> desloca;
    printf("\nO valor do bit eh %d\n",valor_byte);
}
```

```
#include <stdio.h>
int main ()
{
    unsigned char desloca, valor_byte, aux=1;
    printf("\nDigite um numero pertencente ao intervalo [0,255]: ");
    scanf("%hhu", &valor_byte);
    printf("\nDigite o bit a testar(0 a 7): ");
    scanf("%hhu", &desloca);
    aux = aux << desloca;
    valor_byte = valor_byte & aux;
    valor_byte = valor_byte >> desloca;
    printf("\nO valor do bit eh %hhu\n", valor_byte);
}
```



Funções Matemáticas

Operador Condicional

Funções Matemáticas

➤ `abs ()` /*#include <math.h>*/

➤ Calcula o valor absoluto de um inteiro.

➤ Sintaxe: **Tipo de retorno**

int `abs` (`int`);

Parâmetros

Exemplo:

```
...  
int i;  
i = -8;  
printf ("%d", abs(i));  
/*escreve na tela 8*/
```

Funções Matemáticas

➤ *fabs ()* /*#include <math.h>*/

➤ Calcula o valor absoluto de um valor real.

➤ Sintaxe:

```
float fabs (float);
```

Exemplo: ...

```
float f;
```

```
f = -3.7;
```

```
printf ("%f", fabs(f));
```

```
/*escreve na tela 3.70000*/
```


Funções Matemáticas

➤ `pow ()` /*#include <math.h>*/

➤ Potenciação x^y .

➤ Sintaxe:

```
double pow (double x, double y);
```

Exemplo:

```
...
```

```
printf ("%0.2lf", pow(2, 3));
```

```
/* escreve na tela 8.00 */
```

```
printf ("%0.2f", (float)pow(2, 3));
```

```
/* escreve na tela 8.00 */
```

Funções Matemáticas

➤ `sqrt ()` `/*#include <math.h>*/`

➤ Raiz quadrada.

➤ Sintaxe:

```
double sqrt (double);
```

Exemplo: ...

```
printf (“%.3lf”, sqrt(9));  
/* escreve na tela 3.000 */  
printf (“%d”, (int)sqrt(9));  
/* escreve na tela 3 */
```

Funções Matemáticas

➤ `log ()` /*#include <math.h>*/

➤ Logarítmo natural.

➤ Sintaxe:

```
double log (double);
```

Exemplo:

```
...
```

```
printf (“%.1f”, log(30));
```

```
/* escreve na tela 3.4 */
```

Funções Matemáticas

➤ *log10 ()* /*#include <math.h>*/

➤ Logarítmo na base 10.

➤ Sintaxe:

```
double log10 (double);
```

Exemplo:

```
...
```

```
printf (“%.3lf”, log10(30));
```

```
/* escreve na tela 1.477 */
```

Funções Matemáticas

➤ `cos ()` /*#include <math.h>*/

➤ Cosseno.

➤ Sintaxe:

```
double cos (double x); /* x em radianos */
```

Exemplo:

```
...
```

```
printf ("%0.2lf", cos(3.141592/4));
```

```
/* escreve na tela 0.71 */
```

Funções Matemáticas

➤ `sin () /*#include <math.h>*/`

➤ Seno.

➤ Sintaxe:

```
double sin (double x); /* x em radianos */
```

Exemplo:

```
...
```

```
printf ("%0.2lf", sin(3.141592/3));
```

```
/* escreve na tela 0.87 */
```

Funções Matemáticas

➤ `tan ()` /*#include <math.h>*/

➤ Tangente.

➤ Sintaxe:

```
double tan (double x); /* x em radianos */
```

Exemplo:

```
...
```

```
printf (“%.2lf”, tan(3.141592/5));
```

```
/* escreve na tela 0.73 */
```

Exercício

Construa um programa, na linguagem C, que receba através da entrada padrão um valor real para x e avalie a expressão $\sqrt{(3 + x)^3 * 5}$, apresentando o resultado da avaliação na saída padrão.