EXPRESIONES Y OPERADORES

Los distintos operadores permiten formar expresiones tanto aritméticas como lógicas.

PRECEDENCIA DE LOS OPERADORES

PRECEDENCIA	OPERADORES	ASOCIATIVIDAD
0	()[] -> .	izq. a derecha
1	sizeof (tipo) ! ~ ++ -- signo* &	derecha a izq.
2	* / %	izq. a derecha

3	+ -	izq. a derecha
4	>	izq. a derecha
5	>=	izq. a derecha
6	== !=	izq. a derecha
7	&	izq. a derecha
8	^	izq. a derecha
9	|	izq. a derecha
10	&&	izq. a derecha
11	||	izq. a derecha
12	?:	derecha a izq.

13	= += -= *= etc	derecha a izq.

Símbolos Especiales

Hay un grupo de símbolos, que son tratados como caracteres individuales, que especifican algunos caracteres especiales del código ASCII. Los más importantes son:

\a	Alert (bit)
\b	Backspace
\f	Formfeed

\n	Newline
\r	carriage return
\t	horizontal tab
\v	vertical tab
\\	Backslash
\'	single quote
\"	double quote
\OOO	visualiza un carácter cuyo código ASCII es OOO en octal.
\xHHH	visualiza un carácter cuyo código ASCII es HHH en hexadecimal.

Más Formatos (para Datos) para lectura e impresión.

%d, %i	entero decimal con signo
%o	entero octal sin signo
%u	entero decimal sin signo
%x	entero hexadecimal sin signo (en minúsculas)
%X	entero hexadecimal sin signo (en mayúsculas)
%f	coma flotante en la forma
%e	coma flotante en notación científica
%g	usar %e o %f, el más corto
%E	como e pero en mayúsculas

%G	como g pero en mayúsculas
%c	un carácter
%s	cadena de caracteres terminada en '\0'
%%	imprime el carácter %
%p	puntero
%l	entero largo

Librería "math" ANSI C

Contiene los prototipos de las funciones y otras definiciones para el uso y manipulación de funciones matemáticas.

Funciones

acos	asin	atan	atan2	pow	sqrt
ceil	cos	cosh	exp	sin	tan
fabs	floor	fmod	frexp	sinh	
ldexp	log	log10	modf	tanh	

Librería “stdio” ANSI C

Contiene los prototipos de las funciones, macros, y tipos para manipular datos de entrada y salida.

Funciones

clearerr	fclose	feof	ferror	remove	rename
fflush	fgetc	fgetpos	fgets	setbuf	setvbuf
fopen	formato	fprintf	fputc	tmpfile	tmpnam
fputs	fread	freopen	fscanf	vprintf	vsprintf

fseek	fsetpos	ftell	fwrite	rewind	scanf
getc	getchar	gets	perror	sprintf	sscanf
printf	putc	putchar	puts	ungetc	vfprintf

Librería “string” ANSI C

Contiene los prototipos de las funciones y macros de clasificación de caracteres.

Funciones

memchr	memcmp	memcpy	memmove	strpbrk	strrchr
memset	strcat	strchr	strcmp	strspn	strstr
strcoll	strcpy	strcspn	strerror	strtok	strxfrm
strlen	strncat	strncmp	strncpy		

Librería “stdlib” ANSI C

Contiene los prototipos de las funciones, macros, y tipos para utilidades de uso general.

Funciones

abort	abs	atexit	atof	system	wct
atoi	atol	bsearch	calloc	strtod	strtol
div	exit	free	getenv	realloc	srand
labs	ldiv	malloc	mblen	qsort	rand
mbstowcs	mbtowc	strtoul			

Librería “time” ANSI C

Contiene los prototipos de las funciones, macros, y tipos para manipular la hora y la fecha del sistema.

Funciones

asctime	clock	ctime	difftime	time	strftime
gmtime	localtime	mktime			

ARREGLOS UNIDIMENSIONALES

Es la colección de variables del mismo tipo que usa un nombre común. Un arreglo puede tener una o varias dimensiones. Para acceder a un elemento específico de un arreglo se usa su índice.

Formato:

TipoDato nombre[tamaño];

Ej.

int lista[8];

En memoria seria:

Posición	Elemento
0	45
1	6
2	7
3	4
4	98
5	32
6	12
7	3

Ejem1. /*Almacenar en un arreglo de 100 elementos de tipo entero los números del 1, 2… 100, mostrar cada dato al finalizar.*/

```
#include<stdio.h>
int main(){
   int i, a[100];
   for(i=0; i<=99; i++)
     {
          a[i]=i+1;
          printf("%d\t", a[i]);
     } return 0;}
```

Ejem2. /*Almacenar en un arreglo de 100 elementos de tipo entero los números del 100, 99… 1, mostrar cada dato al finalizar.*/

```
#include<stdio.h>

int main(){

   int i, a[100];

   for(i=0; i<=99; i++)

   {

     a[i]=100-i;

     printf("%d\t", a[i]);

   }return 0;}
```

Ejem1a. /*Almacenar en un arreglo de 100 elementos de tipo entero los números del 1, 2… 100, mostrar cada dato al finalizar.*/

```
#include<stdio.h>
int main(){
   int i, a[100];
   for(i=0; i<=99; i++)
     {
          a[i]=i+1;
          printf("posición:%d, contenido:%d\n", i, a[i]);
     }return 0; }
```

Ejem2a. /*Almacenar en un arreglo de 100 elementos de tipo entero los números del 100, 99… 1, mostrar cada dato al finalizar.*/

```
#include<stdio.h>

int main(){

   int i, a[100];

   for(i=0; i<=99; i++)

   {

     a[i]=100-i;

     printf("posición:%d, contenido:%d\n", i, a[i]);

   }return 0;}
```

ARREGLOS BIDIMENSIONALES

La forma más sencilla de un arreglo multidimensional es la de dos dimensiones, es decir, es un arreglo de arreglos.

Formato:

Tipo nombre[tam1][tam2];

Ej.

```
int matriz1[3][3], indice1, indice2;

/*Donde indice1 e indice2 son necesariamente variables
distintas y de tipo entero siempre*/
```

Gráficamente lo veríamos:

Posición

00	01	02
10	11	12
20	21	22

Elementos

2	3	5
3	4	5
45	56	78

Ejem3.

/*Almacenar en un arreglo de 3 X 3 solamente uno´s, imprimir en pantalla la matriz.*/

```
#include<stdio.h>
int main(){
   int i,j,a[3][3];
   for(i=0; i<=2; i++)//ciclo de filas
   {
    for(j=0; j<=2; j++)//ciclo de columnas
    {
        a[i][j]=1;//asignación de contenido de la matriz a
    }
   }
```

```
    for(i=0; i<=2; i++)//impresión en forma de matriz
    {
     for(j=0; j<=2; j++)
     {
         printf("%i\t", a[i][j]);
     }
    printf("\n");
    }
return 0;
}
```

Ejem4.

/* Almacenar en un arreglo de 4 X 4 la numeración del 1, 2… 16, imprimir en pantalla la matriz.*/

```
#include<stdio.h>
int main(){
   int i,j,a[4][4], vAux=1;
   for(i=0; i<=3; i++)
   {
    for(j=0; j<=3; j++)
    {
        a[i][j]=vAux;
        vAux++;
    }   }
```

```
    for(i=0; i<=3; i++)
    {
     for(j=0; j<=3; j++)
     {
          printf("%i\t", a[i][j]);
     }
    printf("\n");
    }
return 0;
}
```