

UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

- FACULTAD DE INGENIERÍA -

CARRERA: Ingeniería en Computación

ALUMNO: Medrano Miranda Daniel Ulises

MATERIA: Estructura de Datos y Algoritmos I

PROFESOR: Marco Antonio Martínez Quintana

ACTIVIDAD ASÍNCRONA LUNES #2

Repaso de C después del Paro

GRUPO: 15

SEMESTRE: 2021 – 2

FECHA: 07 / Junio / 2021

Operadores

Operadores Aritméticos	Operadores Relacionales	Operadores Lógicos
- Resta + Suma * Multiplicación / División real % Residuo = Asignación	< Menor que > Mayor que <= Menor igual que >= Mayor igual que != Desigualdad == Igualdad	&& Y O ! Negación ++ Incremento -- Decremento

Declaración de Variables

Tipo	Declaración	Limite Inferior	Limite Superior
Entero	Int A;	-32768	32767
Entero Corto	Short Int A;	-128	127
Entero Largo	Long Int A;	2E -6	2E 6
Entero sin Signo	Unsigned Int A;	0	65535
Entero con Signo	Signed Int A;	-65000	65000
Real	Float A;	-3.4E37	3.4E 38
Real Doble	Double A;	-1.7E -304	1.7E 308
Real Largo	Long DoubleA;	3.4E -4932	1.1E 4932
Carácter	Char A;	-128	127
Carácter sin signo	Unsigned Char A;		
Carácter con signo	Signed Char A;		
Palabra	Char[] A;		
Valor Nulo	Void	0	0
Arreglo	Int A[N]		

Usos de printf y scanf

printf("Mensaje");	Escribe Mensaje en la pantalla
printf("Mensaje %d",Variable);	Escribe Mensaje y el valor de la Variable en pantalla
printf("Mensaje %d",Variable:2:3);	Escribe Mensaje y el valor de la Variable con 2 enteros y 3 decimales
scanf("%d",&Variable);	Asigna valor entero a Variable
scanf("%d %f",&Variable1,&Variable2);	Asigna valor entero a Variable1 y valor real a Variable2

Formato para Variable

Formato de Barra Invertida

Formato	Descripción	Formato	Descripción
%c	Un carácter	\b	Retroceso
%d	Real o Entero	\f	Salto de línea
%i	Real	\n	Nueva línea
%e	Notación con "e"	\r	Retorno de carro
%E	Notación con "E"	\t	Tabulador horizontal
%f	Real	\"	Comillas
%g	Real	\'	Apóstrofo
%G	Real con notación "E"	\n	Carácter nulo
%s	Cadena	\\	Barra invertida
%u	Real sin signo	\v	Tabulador Vertical
%%	Imprime %	\a	Campanilla

Sentencias de Control

<pre>for(Contador = 1; Contador <= N; Contador++) { Sentencia; }</pre>	De Contador=1 hasta Contador<=N Incremento
<pre>if(Condición == 1) { Sentencia; } else { Sentencia; }</pre>	Si Condición = 1 Entonces Si no
<pre>while(Condición == 1) { Sentencia; }</pre>	Mientras Condición = 1 haz
<pre>do{ Sentencia; }(Condición == 1);</pre>	Haz Hasta Condición = 1
<pre>switch(Opción) { 1: Sentencia1; break; 2: Sentencia2; break; default: Sentencia3; break; }</pre>	En caso de Opción Si Opción = 1 Sentencia1 rompe Si Opción = 2 Sentencia2 rompe Si no Sentencia3 rompe

ARREGLOS

Los arreglos (arrays) permiten almacenar vectores y matrices. Los arreglos unidimensionales sirven para manejar vectores y los arreglos bidimensionales para matrices. Sin embargo, las matrices también se pueden almacenar mediante arreglos unidimensionales y por medio de apuntadores a apuntadores, temas que se verán en el capítulo siguiente.

La palabra unidimensional no indica que se trata de vectores en espacios de dimensión uno; indica que su manejo se hace mediante un subíndice. El manejo de los arreglos bidimensionales se hace mediante dos subíndices.

Arreglos unidimensionales

El siguiente ejemplo muestra la definición de tres arreglos, uno de 80 elementos doble precisión, otro de 30 elementos enteros y uno de 20 elementos tipo carácter.

```
double x[80];
```

```
int factores[30];
```

```
char datos[20];
```

Arreglos multidimensionales

La declaración de los arreglos bidimensionales, caso particular de los arreglos multidimensionales, se hace como en el siguiente ejemplo:

```
double a[3][4];
```

```
int pos[10][40];
```

```
char list[25][25];
```