

PRINCIPIOS DE PROGRAMACIÓN:

TEMA 1:

CONCEPTOS BÁSICOS Y METODOLOGÍA PARA RESOLVER PROBLEMAS

PROGRAMA:

Secuencia de pasos a lógicos para resolver un problema.

ESTRUCTURA:

- **ENTRADA:** Tomar datos de un dispositivo externo (Teclado, Mouse) y dejarlos en memoria.
- **PROCESO:** A los datos dejados en memoria se les manda a la ALU (Unidad Aritmético Lógica) y los devuelve a la memoria.
- **SALIDA:** Se envían a un dispositivo externo y se presenta como información después de ser procesados. (Monitor, Impresora)

LENGUAJE DE PROGRAMACIÓN:

Conjunto de caracteres que nos permiten crear instrucciones siguiendo una sintaxis.

LENGUAJE ALGORÍTMICO:

Orientado a procedimientos y diseñado para ayudar al programador en el diseño y desarrollo de algoritmos.

METODOLOGÍA:

- Entender el problema.
- Hacer un análisis:
 - ¿Qué tenemos?
 - ¿Qué necesitamos?
 - ¿Qué buscamos?
- Diseñar el algoritmo de solución.
- Codificar.

TIPOS DE DATOS:

ENTEROS: (Int) Números enteros positivos o negativos.

REALES: (Flota) Números con decimales.

CARACTERES: (Char) Símbolos, N°, caracteres solos. (@, #, \$, %)

CADENA DE CARACTERES: (String) Agrupación de caracteres.

BOLEANOS: (Boolean) .T. (True), .F. (false)

VARIABLES:

Es un conjunto de símbolos o solo uno que reserva espacio en la memoria y su valor puede cambiar durante la ejecución del programa. Solo números y letras. Números solos NO. Letras solas SI.

CONSTANTES:

No cambia su valor durante la ejecución del programa. Solo números y letras. Números solos NO. Letras solas SI.

SINTAXIS PARA DECLARAR VARIABLES:

INICIAR VARIABLE:

VARIAS VARIABLES:

DECLARAR CONSTANTES:

ASIGNACIÓN:

Aquí le asignamos un valor a la variable a (5), de lo cual se deduce que el valor de dicha variable es 5.

COMPARACIÓN:

Aquí lo que queremos decir es que la variable a es igual al valor 5.

OPERADOR:

Es un símbolo o palabra que nos ayuda a realizar una operación.

Los operadores pueden ser:

ARITMÉTICOS:

SUMA	+
RESTA	-
MULTIPLICACIÓN	*
DIVISIÓN	/
EXPONENTE	, * *
RESIDUO	MOD

RELACIONALES:

MAYOR QUE	>
MENOR QUE	<
MAYOR O IGUAL QUE	>=
MENOR O IGUAL QUE	<=
IGUAL A	==

DIFERENTE A	=
-------------	---

LÓGICOS:

UNIÓN	AND	Los dos deben cumplirse.
INTERSECCIÓN	OR	Con uno que se cumpla.
NEGACIÓN	NOT	Cambia true por false

AND: (Los deben de cumplirse)

A	B	A AND B
0	0	0 (Falso)
0	1	0 (Falso)
1	0	0 (Falso)
1	1	1 (Verdadero)

OR: (Con uno que se cumpla)

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NOT: (Cambia verdadero por falso y viceversa)

OPERANDOS:

Valores o variables que nos permiten presentar un resultado o un dato:

EXPRESIONES.

Conjuntos de operadores y operandos relacionados entre si, con la finalidad de construir una operación valida.

EJERCICIOS:

Resuelve los siguientes 3 ejercicios:

1)

Int a = 4;

Float b = 5.5;

a + b =

a > b =

b = = 8 =

b >= 100 =

100 / 4 =

55 MOD 8 =

30 ** 2 =

400 * 2 =

b |= 5.5 =

b = = 5.5 =

50 MOD 10 =

40 MOD 200 =

2)

b = 8;

b + 4 =

3)

a = 200;

a / 100 =

a Mod 5 =

PRECEDENCIA DE LOS OPERADORES:

NIVEL	OPERADOR
1	, NOT
2	, /, MOD, AND
3	+, -, OR
4	>, >=, <, <=, =, =, =

TÉCNICAS O HERRAMIENTAS PARA LA FORMULACIÓN DE ALGORITMOS:

- **DIAGRAMA DE FLUJO**
- **PSEUDOCÓDIGO.**
- **NASSI-SHNEIDERMAN. (N-S). (ESTRUCTURADO).**

DIAGRAMA DE FLUJO:

Es una representación visual o gráfica del flujo de los datos en la que cada paso de un algoritmo es representado por un símbolo estándar.

PRINCIPALES SÍMBOLOS DEL DIAGRAMA DE FLUJO:

NO SI

NO

SI

ESTRUCTURA BÁSICA DE UN DIAGRAMA DE FLUJO:

SEUDO CÓDIGO:

Es un lenguaje similar al español e inglés, pero más sencillo, donde se utilizan unas series de palabras como instrucciones.

LEER	READ
ESCRIBIR	WRITE
INICIO	
FIN	
IF (Decisión)	
END IF	

EJEMPLOS:

1.- Pedir dos números y presentar la suma:

Diagrama de flujo:

2.- Pedir dos números y presentar la suma:

Seudo código:

INICIO

int N1, N2, RES

ESCRIBIR Cual es el número 1

LEER N1

ESCRIBIR Cual es el número 2

LEER N2

RES = N1 + N2

ESCRIBIR La suma es: RES

FIN

3.- Calcular el área de un triángulo si conocemos que :

INICIO

int A, b, h

ESCRIBIR Cual es la base

LEER b

ESCRIBIR Cual es la altura

LEER h

ESCRIBIR El área es: A

FIN

LENGUAJE DE PROGRAMACIÓN:

Es una notación que permite escribir programas a través de los cuales podemos comunicarnos con la computadora siguiendo normas de sintaxis.

CARACTERÍSTICAS GENERALES DE LENGUAJE:

- C tiene la funcionalidad de un lenguaje de medio nivel, pero reúne las características de un lenguaje de alto nivel.
- Toda palabra clave o instrucción es en minúsculas.
- Una palabra reservada no puede ser usada con otro propósito dentro de un programa.
- Todo programa en C consta por lo menos de una función (main ()) y requiere de archivos de cabecera (conio.h, stdio.h)

ESTRUCTURA O FORMATO DE UN PROGRAMA EN C:

incluye <prototipo> // Archivos de cabecera (h.) Librerías o prototipos.

define // Declaraciones de constantes globales.

// declaraciones y funciones // Declaración de variables y funciones globales.

void main () // Función principal de C.

{ // Inicio del programa principal.

// declaraciones; Cuerpo del programa principal.

```
// sentencias;    .

} // fin del main o programa principal.

funciones.

{

// declaraciones;

// sentencias;

}
```

PROTOTIPOS O LIBRERIAS DE C: (Archivos .h)

NOMBRE	FUNCIÓN RELACIONADA
alloc.h	Asignación dinámica de memoria.
conio.h **	E/S por consola.
ctype.h	Funciones relacionadas con caracteres.
graphics.h	Gráficos.
math.h	Exclusivo para funciones matemáticas.
stdio.h **	E/S estándar.
string.h	Funciones relacionadas con cadenas.
stdlib.h	Funciones variadas.

**** Siempre deben de ir al principio del programa SIEMPRE.**

IDENTIFICADORES:

Todas las variables, constantes, nombres de funciones y cualquier otro objeto definido por el usuario.

REGLAS PARA DEFINIR IDENTIFICADORES:

- Pueden constar de 1 a 255 caracteres.
- Pueden comenzar con letra ó _.
- Después de la primera letra pueden ir números.
- No pueden ser una palabra clave del lenguaje ni tampoco el nombre de una función.
- No pueden llevar caracteres especiales.
- No se pueden usar espacios.

Ejemplos:

- **1x** (Mal).
- **Suma** (Bien).
- **Main** (Mal).
- **Suma\$** (Mal).
- **Suma resta** (Mal).
- **A** (Bien).

TIPOS DE DATOS:

Conjunto de valores con características similares que permiten ser asignados a un identificador.

NOMBRE	TAMAÑO EN BITS	RANGO
CHAR (Letras/Carácter)	8	-128 a 127
INT (Enteros)	16	-32768 a 32767
FLOAT (Decimales)	32	- 34*10 ⁻³⁸ a 34*10 ³⁸
DOUBLE	64	-1.7*10 ⁻³⁰⁸ a 1.7*10 ³⁰⁸
VOID	0	SIN VALOR

MODIFICADORES:

- **SHORT** (Corto)
- **LONG** (Largo)
- **UNSIGNED** (Sin signo)
- **SIGNED** (Con signo)

VARIABLES:

Es un identificador que reserva un espacio en memoria y su valor puede cambiar durante la ejecución del programa.

incluye <prototipo>

var1, var2 varN // Tipo de datos globales.

main ()

{

var1, var2 varN // Tipo de datos locales.

}

CONSTANTES:

Es un identificador que reserva espacio en la memoria y su valor no puede cambiar durante la ejecución del programa.

incluye <prototipo>

const //Tipo de datos globales.

main ()

{

const // Tipo de datos locales.

}

Ejemplos:

const float = const a = 31.908

const int = const b = 23

OPERADORES:

Es un símbolo que indica al compilador que operación debe realizar.

ARITMÉTICOS:

RESTA	-
SUMA	+
MULTIPLICACIÓN	*
DIVISIÓN	/
MODULO o RESTO	%
INCREMENTO DE 1	+ +
INCREMENTO DE 5	a +
DECREMENTO DE 1	- -

RELACIONALES:

MAYOR QUE	>
MENOR QUE	<
MAYOR o IGUAL QUE	>=
MENOR o IGUAL QUE	<=
DIFERENTE	!=

LÓGICOS:

UNIÓN	&&	Y
INTERSECCIÓN		O
NEGACIÓN	!	

A NIVEL BIT:

AND	&
OR	
COMPLEMENTO A 1	
DESPLAZAMIENTO DERECHA	>>
DESPLA. IZQUIERDA	<<

FUNCIONES DE E/S ESTÁNDAR Y POR CONSOLAS:

FUNCIONES E/S ESTÁNDAR: <stdio.h>

printf:

Es la función utilizada para desplegar información en la pantalla.

Sintaxis:

printf(Cadena, <espec>, <espec>);

Ejemplo:

printf(Salvador);

COMANDOS DE FORMATO:

ESPECIFICADOR	FUNCIÓN RELACIONADA
% d	ENTERO.
% f	FLOTANTE --- DECIMALES.
% le ó % lf	DOUBLE.
% e	FLOTANTE CON V. EXPONENCIAL.
% c	CARÁCTER (1).
% s	CADENA DE CARACTERES.

scanf:

Toma la entrada del teclado y la almacena en variables previamente declaradas.

Sintaxis:

scanf(Especificador, &var);

Ejemplo:

scanf(%d, &x); NOTA: Cuando lleva una cadena no se utiliza &.

FUNCIONES E/S POR CONSOLA: <conio.h>

clrscr ();

Limpia la pantalla y posiciona el cursor en la parte superior izquierda.

Cursor

Sintaxis:

clrscr ();

gotoxy (a, b);

Posiciona el cursor en la posición establecida por la columna (a) y renglón (b). 80columnas, 24 renglones.

Renglón 24, Columna 40.

Sintaxis:

gotoxy (40,24);

printf(Salva);

getch ();

Toma un carácter del teclado y no lo despliega en pantalla.

Sintaxis:

getch ();

getche ();

Toma un carácter del teclado y lo despliega en pantalla.

Sintaxis:

getche ();

EJEMPLOS DE PROGRAMAS:

1.- Diseña un programa que realice la suma de dos números, un int y un float, y que muestre el resultado:

include <stdio.h>

include <conio.h>

void main ()

{

clrscr ();

int n1;

float n2, res;

printf(Dame el número 1);

scanf(%d, & n1);

printf(Dame el número 2);

scanf(%f, & n2);

res = n1 + n2;

printf(Resultado %f, res);

getch ();

```
}
```

2.- Calcular el área de un círculo si conocemos que es igual a πr^2 . El valor de Pi deberá ser tomado como constante.

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

```
# include <conio.h>
```

```
# include <math.h>
```

```
void main ( )
```

```
{
```

```
clrscr ( );
```

```
const float Pi = 3.1416;
```

```
int radio;
```

```
float res;
```

```
printf(Dame el valor del radio);
```

```
scanf(%d, & radio);
```

```
res = Pi * (pow (radio, 2));
```

```
printf(El area es: &f, res);
```

```
getch ( );
```

```
}
```

3.- Calcular el sueldo neto de un trabajador en función de horas trabajadas y precio por hora. El sueldo bruto obtenido determina una retención del 8% del seguro de vida, que es en base al sueldo bruto, presente el sueldo neto y la retención del seguro.

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

```
# include <conio.h>
```

```
void main ( )
```

```
{
```

```
clrscr ( );
```

```
int horas, precio,
```

```
float ret, sb, sn;
```

```

printf(Horas trabajadas);

scanf(%d, & horas);

printf(Precio horas);

scanf(%d, & horas);

sb = precio * horas;

ret = sb * 0.08;

sn = sb - ret;

printf(La retención es:%f, ret);

printf(Su sueldo neto es:%f, sn);

getch ( );

}

```

4.- Diseñe un programa que lea un número y determine la raíz cuadrada del mismo, presente el resultado:

```

# include <stdio.h>

# include <conio.h>

# include <math.h>

void main ( )

{

clrscr ( );

float N, res;

printf(Dame un número);

scanf(%f, & N);

res = sqrt(N);

printf(La raíz cuadrada es:%f, res);

getch( );

}

```

SENTENCIAS CONDICIONALES:

IF:

Permite elegir entre opciones de ejecución.

1.-

if (expresión) .T.

sentencia;

2.-

if (expresión) .T.

sentencia;

else

sentencia;

3.-

if (expresión)

{

sentencia1;

sentenciaN;

}

4.-

if (expresión)

{

sentencia1;

sentenciaN;

}

```

else

{

sentencia1;

-----

-----

sentenciaN;

}

```

EJEMPLOS DE PROGRAMAS CON IF:

1.- Leer un número y determine si es par o impar, presente un mensaje:

```

# include <stdio.h>

# include <conio.h>

void main ( )

{

clrscr ( );

int num;

printf(Inserte el número);

scanf(%d, & num);

if (num % 2 == 0)

printf(El número es par);

else

print(El número es impar);

getch ( );

}

```

2.- Determine la cuenta de un cliente si sabemos que se consumieron una sopa a 8 pesos y un jugo a 4 pesos. Si la cuenta es superior a 30 pesos aplíquese un descuento del 10%. Presente la cuenta.

```

# include <stdio.h>

# include <conio.h>

```

```

void main ( )

{

clrscr ( );

int S, J;

float Desc, T, X;

printf(Cuántas sopas consumió);

scanf(%d, & S);

printf(Cuántos jugos consumió);

scanf(%d, & J);

T = (S*8) + (J*4);

if (T > 30)

{

X = T * 0.10;

Desc = T - X;

printf(La cuenta es:%d, Desc);

}

else

{

printf(La cuenta es:%f, T);

}

getch ( );

}

```

IF ANIDADOS:

1.- Ejemplo:

```
if (Expresión)
```

```
if (Expresión)
```

sentencia;

else

sentencia;

2.- Ejemplo:

if (Expresión)

if (Expresión)

{

sentencia1;

sentenciaN;

}

else

sentencia;

**** NOTA: Un Else le pertenece al IF inmediato que tenga arriba.**

EJEMPLO DE IF ANIDADOS:

1.- Leer tres números y determinar cual es el mayor de ellos:

include <stdio.h>

include <conio.h>

void main ()

{

clrscr ();

int N1, N2, N3;

printf(Inserte el primer número);

scanf(%d, & N1);

printf(Inserte el segundo número);

```

scanf(%d, & N2);

printf(Inserte el tercer número);

scanf(%d, & N3);

if (N1 > N2)

if (N1 > N3)

printf(El mayor es N1);

else

printf(El mayor es N3);

if (N2 > N3)

printf(El mayor es N2);

else

printf(El mayor es N3);

getch ( );

}

```

IF ESCALONADOS:

```

if (Expresión)

sentencia;

else

if (Expresión)

sentencia;

else

if (Expresión)

sentencia;

else

sentencia;

```

**** NOTAS:**

toupper = Convierte a mayúsculas.

tolower = Convierte a minúsculas.

char op;

op = Toupper (getche());

op = tolower (getche());

EJEMPLO DE IF ESCALONADO EN UN PROGRAMA:

1.- Diseña un programa que presente el siguiente menú: a) Suma, b) Resta, c) Multiplicación, d) Raíz Cuadrada. Realice la operación y presente el resultado.

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

```
# include <conio.h>
```

```
# include <math.h>
```

```
# include <ctype.h>
```

```
void main ( )
```

```
{
```

```
clrscr ( );
```

```
int n1, n2, n3;
```

```
float Res;
```

```
char op;
```

```
printf(a)Suma);
```

```
printf(b)Resta);
```

```
printf(c)Multiplicación);
```

```
printf(d)Raíz Cuadrada);
```

```
printf(Cual es tu opción);
```

```
op = tolower(getche ( ));
```

```
if (op == `a' || op == `b' || op == `c')
```

```
{
```

```
printf(Inserte el número 1);
```

```

scanf(%d, & n1);

printf(Inserte el número 2);

scanf(%d, & n2);

}

if (op == 'd')
{
printf(Inserte el número 3);

scanf(%d, & n3);

}

if (op == 'a')

printf(Suma %d, n1 + n2);

if (op == `b')

printf(Resta %d, n1 - n2);

if (op == `c')

printf(Multiplicación %d, n1 * n2);

if (op == `d')
{
Res = sqrt(n3);

printf(Raíz %f, res);

}

else

printf(Error);

getch ( );

}

int a;

float b;

```

int a = 10;

int a, b, c, d, e;

const int a = 5

const + tipo dato + nombre constante = valor;

const float Pi = 3.1416

a = 5

a == 5

1 = 0

0 = 1

SUELDO > 100

operandos

SUELDO > 100 = Expresión

INICIO / FIN PANTALLA

E/S---I/O CONECTOR

PÁGINA OUT

PROCESO CONECTOR

PÁGINA IN

LEER / TECLEAR

DECISIÓN

ITERACIÓN DO WHILE

DECISIÓN MÚLTIPLE

ITERACIÓN FOR

INICIO

ACCIÓN

1

ACCIÓN

2

ACCIÓN

3

ACCIÓN

N

FIN

INICIO

INT N1, N2, RES

QUAL ES EL

NUMERO

1

N 1

QUAL ES EL

NUMERO

2

N 2

$RES = N1 + N2$

LA SUMA

ES:

RES

FIN

Salva

.F.

.T.