CONSIDERACIONES GENERALES PARA CODIFICAR EN EL LENGUAJE DE PROGRAMACIÓN C

Bibliotecas del lenguaje

Formato general**: #include <nombre de biblioteca.h>**

Ejemplo:

#include <stdio.h> <-Funciones de entrada y salida de datos.

#include <conio.h> <-Funciones de consola.

#include <math.h> <-Funciones de matemática.

#include <stdlib.h> <-Funciones del estándar ANSI C.

#include <ctype.h> <-Funciones de conversión de datos.

Comentarios

```
//Comentario de una sola línea.

/*
 Comentario
 De más de una
 Línea o para una
sola línea.
*/
```

Sentencias

Terminan con PUNTO Y COMA (;) a excepción de las directivas al pre compilador o preprocesador (#include, #define), las estructuras de: selección, iteración (excepto do-while) y los encabezados de funciones en la sección del desarrollo de las mismas.

Nomenclaturas de variables o identificadores

- El primer caracter debe ser una letra o guión bajo.
- No pueden contener caracteres especiales o espacio.
- No puede contener la letra ñ o Ñ.
- No se puede colocar un nombre que sea igual a una palabra reservada del lenguaje.
- No se puede colocar un nombre que sea igual a una función del lenguaje o una función desarrollada por el programador.
- Letra MAYÚSCULA distinto de letra MINÚSCULA (sensitivo).

Palabras reservadas

include, define, struct, int, float, char, if,void, switch, for,while, do, goto, continue, break, return, enum, union, goto, auto, static.

Declaración de variables

Formato general**: <tipo de dato> <identificador o nombre>;**

<tipo de dato>:	**int**	**float**	**char**
<formato>:	**%d**	**%f**	**%c**

Ejemplo: Solo para declarar.

int Variable;

float x;

char car;

Ejemplo para declarar una lista de variables del mismo tipo:

int var1, var2, var3, var4, var5;

Ejemplo: declarar e inicializar variables.

int Variable = 1;

float x = 2.5;

char car = 'c';

Declaración e inicialización de constantes

Por lo general el nombre o identificador se indican con mayúsculas

1. **Como etiqueta o label (reemplazo directo, no ocupa espacio de la memoria principal)**

   Formato general**: #define <Nombre de la constante> <valor>**

Ejemplo:

#define LETRA 'c' //Constante de tipo char se coloca entre apostrofes simple.

```
#define PI 3.14159 //Caracter punto es el separador de decimales.
#define TAM 10
```

2. **Como una variable (ocupa espacio de la memoria principal)**

Formato general: const <tipo de dato> <identificador> = <valor>;

Ejemplo:
const int TAM = 10;

Llamado o invocar a funciones

Formato general: <Nombre de la función> (<Parámetro/s>);

<parámetro>: Es/son el/los valor/es que se le envía/n a la función. Puede ser nada (vacío) o **void**, uno o varios. El carácter coma es un separador de parámetros.

Ejemplo:

```
getch();              //genera una pausa, no es estándar -Función sin parámetros-.
system("PAUSE");      //genera una pausa, es estándar -Función con un parámetro-.
fflush(stdin);        //Limpia el buffer de entrada -Función con un parámetro- .
```

Función printf (Exhibir mensajes en la pantalla o monitor).

Formato general: printf ("cadena de control", argumentos);

- **Cadena de control:** Texto literal que se exhibe en la pantalla y/ó formato de variable.
- **Argumentos:** Puede ser una variable/s u operación/es que se relacionan con el formato indicado en la cadena de control. El primer formato indicado se relaciona con el primer argumento, el segundo formato se relaciona con el segundo argumento, etc.

Ejemplo para exhibir un mensaje o leyenda

```
printf ("Primer Programa");
```

Ejemplo para exhibir un mensaje y el valor de una variable

```
Int a=5;

printf ("El valor de la variable a es %d",a);
```

Ejemplo para exhibir un mensaje y el valor de una variable y una operación

```
Int a=5,b=1;

float x=2.5;

printf ("El valor de la variable a es %d, la suma de a y b es %d la variable x vale %.2f",a, (a+b),x); //%.2f
        muestra dos dígitos decimales.
```

Función scanf (Almacenar lo ingresado por teclado en las variables- Entrada o ingreso).

Formato general**: scanf ("cadena de control", & <nombre de variable>);**

- **Cadena de control:** Formato de variable (%d, %f, %c).
- **& =** Dirección de memoria**.**

Ejemplo para leer una variable

int a;

printf("Ingrese un valor entero:");

scanf("%d",&a);

Ejemplo para leer más de una variable en la misma línea (solo para tipo int,float).

int a;

float b;

printf("Ingrese un valor entero y una valor real separado por espacio:");

scanf("%d%f",&a,&b);

Ejemplo para leer una variable de tipo char.

int x;

char car;

printf("Ingrese un valor entero");

scanf("%d",&x);

printf("Ingrese un caracter");

fflush(stdin); /*Limpiar el caracter enter del buffer del teclado.

Debe estar antes del scanf, es solo para variables de tipo **char***/

scanf("%c",&car);

Código ASCII – Combinación de Teclas

Caracter	Combinación de Teclas
\	Alt + 92
|	Alt + 124
{	Alt + 123
}	Alt + 124
(	Alt + 40
)	Alt + 41
[	Alt + 91
]	Alt + 93
<	Alt + 60
>	Alt + 62

“	**Alt + 34**

Caracteres válidos

= + - * / % ! ( ) [ ] { } “ ‘ < > & | # , . ;

Operador	Función	Tipo
=	Asignación	Binario
+	Suma	Binario
-	Resta	Binario
*	Multiplicación	Binario
/	División	Binario
%	Resto (De la división entera)	Binario
&	Dirección de memoria	Unario
!	Negación (NOT)	Unario
<	Comparador de menor	Binario
>	Comparador de mayor	Binario

Operadores Combinados	Función	Tipo
==	Comparación por igualdad	Binario
!=	Comparación por desigualdad	Binario
&&	Operador lógico AND	Binario
||	Operador lógico OR	Binario
+=	Acumulador de la suma	Binario
<=	Comparador por menor e igual	Binario
>=	Comparador por mayor e igual	Binario
-=	Acumulador de la resta	Binario
*=	Acumulador del producto	Binario
/=	Acumulador de la división	Binario
--	Post o pre decremento	Unario
++	Post o pre incremento	Unario

Símbolo	Función
,	Separador de parámetros o lista
.	Separador de dígitos decimales
{	Apertura de un bloque (Función, estructura, etc.)
}	Finalización de un bloque (Función, estructura, etc.)
Símbolos combinados	**Función**
//	Comentario de una línea
/*	Inicio del Comentario para más de una línea o una línea.
*/	Fin del comentario de más de una línea o de una línea que comienza con */