

Lenguaje De Programación C





Introducción

- El lenguaje de programación en C, es un lenguaje conocido como de alto nivel.
- Es un lenguaje estructurado, lo que permite generar código claro y sencillo, ya que esta basado en la modularidad.
- C ha tenido un gran impacto en el diseño de otros muchos lenguajes. Ha sido, por ejemplo, la base para definir la sintaxis y aspectos de la semántica de lenguajes tan populares como Java y C++.

C es un lenguaje compilado: antes de ejecutar un programa escrito por nosotros, suministramos su código fuente a un compilador de C. También hay Intérpretes en C.

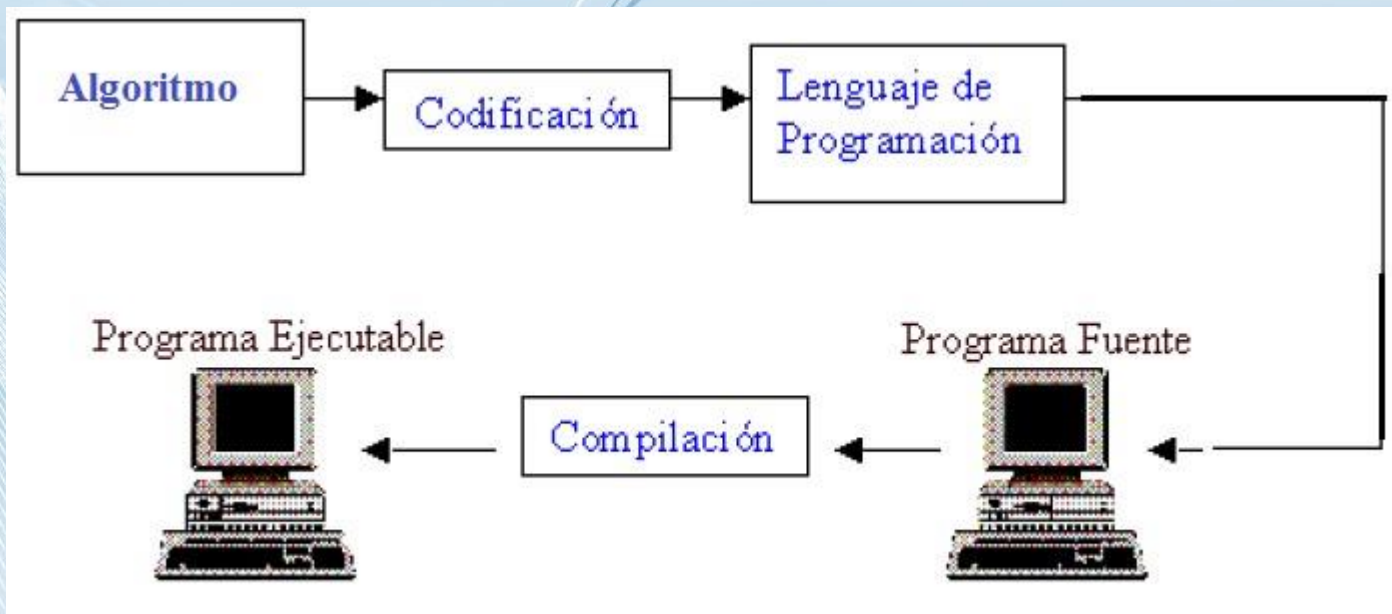
El compilador lee y analiza todo el programa. Si el programa está correctamente escrito según la definición del lenguaje, el compilador genera un nuevo fichero con su traducción a código de máquina, y si no, muestra los errores que ha detectado.

Para ejecutar el programa utilizamos el nombre del fichero generado. Si no modificamos el código fuente, no hace falta que lo compilemos nuevamente para volver a ejecutar el programa: basta con volver a ejecutar el fichero generado por el compilador

Nombre	Fecha de modifica...	Tipo	Tamaño
 EjerIF2	09/03/2017 8:57	CPP - Fuente C++	1 KB
 EjerIF2	09/03/2017 8:57	Aplicación	49 KB



- **Programa o código fuente:** Programa escrito en algún lenguaje y que no ha sido traducido a lenguaje de máquina.
- **Programa o código objeto:** Programa que ya se encuentra en lenguaje de máquina.
- **Compilador:** Traduce todo el programa y genera un código listo para funcionar



Enlace a la página para descargar el programa C



<http://zinjai.sourceforge.net/>



Palabras Reservadas

Son aquel grupo de identificadores (palabras) que no pueden ser utilizadas por el usuario para nombrar a las variables, funciones, procedimientos, objetos y demás elementos de programación que cree.

Ejemplos de palabras reservadas en C:

If – char – int –long –void -- printf

- **Una variable o una constante no pueden llamarse igual que una palabra reservada**
- **Es preciso insistir en que C hace distinción entre mayúsculas y minúsculas. Por lo tanto, la palabra reservada for no puede escribirse como FOR, pues el compilador no la reconoce como una instrucción, sino que la interpreta como un nombre de variable.**



Identificadores

Un identificador es un conjunto de caracteres alfanuméricos que sirve para identificar las entidades del programa (clases, funciones, variables, etc.)

Los identificadores pueden ser combinaciones de letras y números.

Cada lenguaje tiene sus propias reglas que definen como pueden estar contruidos

Ejemplos de identificadores válidos son:

Precio_Venta

Num1

_123

D_i_5

No son válidos:

Precio Venta Lleva un espacio en blanco

1Num Empieza por un número

Precio-Venta Lleva un guión

Identificadores



En C, se debe tener en cuenta que todo identificador debe cumplir las siguientes reglas de sintaxis:

- **Consta de uno o más caracteres.**
- **El primer carácter debe ser una letra o el carácter *subrayado* (_), mientras que, todos los demás pueden ser letras, dígitos o el carácter *subrayado* (_). Las *letras* pueden ser minúsculas o mayúsculas del alfabeto inglés. Así pues, no está permitido el uso de las letras 'ñ' y 'Ñ'.**
- **No pueden existir dos identificadores iguales, es decir, dos elementos de un programa no pueden nombrarse de la misma forma. Lo cual no quiere decir que un identificador no pueda aparecer más de una vez en un programa.**
- **No debe haber un espacio en blanco entre las letras o palabras que conforman el identificador.**
- **De un identificador sólo son significativos los 32 primeros caracteres.**

TP – 1 -----Ejercicio N° 1:

Marcar con una cruz los identificadores válidos. Si no son válidos, explicar por qué.

Identificador	Validos	Explicación
registro1		
1registro		
float		
Nombre y dirección		
e-mail		
e_mail		
\$impuesto		
1234		

Tipos de Datos

char

- Es capaz de contener un carácter

int

- Define un valor numérico entero
- No acepta fracción.

float

- Punto flotante de precisión normal
- Puede ser una fracción o un entero con exponente

double

- Punto Flotante de precisión doble
- Mas cifras significativas para la fracción .



Constantes

Las constantes son valores que no pueden cambiar en la ejecución del programa.

Las constantes de carácter van encerradas en comillas simples. Las constantes enteras se especifican con números sin parte decimal y las de coma flotante con su parte entera separada por un punto de su parte decimal.



Variables

Las variables son valores que se pueden modificar durante la ejecución de un programa.

Declaración

Una declaración asocia un tipo de datos especificado a un grupo de variable.

- Se deben declarar todas las variables antes de que aparezcan en las instrucciones ejecutables.
- Una declaración consta de un tipo de datos, seguido de uno o más nombres de variables, finalizando con un punto y coma.



Ejemplo de Declaración

```
int a, b, c;  
float raiz1, raiz2;  
char indicador, texto[80];
```

De esta forma se declaran a, b y c como variables enteras, raiz1 y raiz2 son variables en coma flotante, indicador una variable de tipo carácter y texto un array de tipo carácter de 80 elementos. Observe los corchetes que delimitan la especificación de tamaño de texto.

También se podrían haber escrito las declaraciones anteriores como sigue:

```
int a;  
int b;  
int c;  
float raiz1;  
float raiz2;  
char indicador;  
char texto[80];
```



Ejemplo de Declaración

Ejemplos de declaración de variables:

`int a;` *Se reserva un espacio en la memoria llamado "a", con capacidad para un entero.*

`float b,c,d;` *Se reservan 3 espacios en la memoria para guardar 3 números reales, a los cuales se hace referencia mediante "b", "c" y "d" respectivamente.*

`char j;` *Se reserva un espacio en la memoria para poder almacenar cualquier carácter y se puede hacer referencia a este espacio mediante el identificador "j".*



Comentario

Son cadenas de caracteres o texto que describen partes del programa que el programador desea explicar.

Dicho texto no forma parte del programa fuente sino una descripción del mismo.

Para poder usar comentarios en un programa C y que el compilador no lo considere como instrucciones del programa se utilizan los símbolos para abrir el texto `/*` y para cerrar el comentario `*/`.

También se puede usar `//`

Ejemplo:

```
/* este es un programa */  
//este es un programa
```

Ejercicio N° 2

Completar la siguiente tabla indicando para las siguientes opciones cuales representan la declaración de una constante:

Opciones	Declaración de Constante
<code>const char c='y'</code>	
<code>t=4</code> ¿se puede escribir de otras formas? Si es así, ¿Cuáles pueden ser?	
<code>int a</code>	



Librería Estándar de C

La librería estándar de C es una recopilación de ficheros cabecera y librerías con rutinas que implementan operaciones comunes, como las de entrada salida o el tratamiento de cadenas.

- El nombre y las características de cada función se encuentran en un fichero denominado archivo de cabecera (con extensión ".h"), pero la implementación real de las funciones están separadas en un archivo de la librería.
- Estas librerías se deben colocar en un programa de lenguaje programación en C, en la instrucción o directiva conocida como **include**.



Ejemplos de Librería Estándar en C

<ctype.h>	Contiene funciones para clasificar caracteres según sus tipos o para convertir ente mayúsculas y minúsculas independientemente del conjunto de caracteres (típicamente ASCII o alguna de sus extensiones).
<math.h>	Contiene las funciones matemáticas comunes.
<stdio.h>	Proporciona el núcleo de las capacidades de entrada/salida del lenguaje C (incluye la venerable función printf).
<string.h>	Para manipulación de cadenas de caracteres.
<time.h>	Para tratamiento y conversión entre formatos de fecha y hora.

Estructura básica de un Programa en Lenguaje C

```
/* Descripción de Bibliotecas*/
```

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

```
/* Función Principal*/
```

```
int main ( )
```

```
{
```

```
/* Cuerpo del programa*/
```

```
printf("Hola mundo \n");
```

```
return;
```

```
}
```



Estructura de un Programa en C

#include <stdio.h>

- **#INCLUDE ES LO QUE SE LLAMA UNA DIRECTIVA.** Sirve para indicar al compilador que incluya otro archivo. Cuando en compilador se encuentra con esta directiva la sustituye por el archivo indicado. En este caso es el archivo **stdio.h** que es donde está definida la función **printf**.
- La directiva, le dice a C que en el proceso de compilación incluya un archivo denominado **stdio.h**. Este fichero se suministra como parte del compilador de Turbo C y contiene la información necesaria para el correcto funcionamiento de la E/S de datos.
- La sentencia **#include** no es una instrucción C. El símbolo **#** la identifica como una **directiva**.
- Los archivo ***.h** se denominan **archivos de cabecera**. Todos los programas C requieren la inclusión de uno o varios archivos de este tipo, por lo que normalmente es necesario utilizar varias líneas **#include**.

Estructura de un Programa en C

El lenguaje C proporciona fundamentalmente dos formas de manejo de funciones: las internas y las externas.

Las funciones internas son las ya implementadas e incorporadas en el lenguaje de programación. Para poder hacer uso de ellas, es necesario incluir el archivo de cabecera estándar correspondiente (también llamada biblioteca) al que corresponda cada función. La forma de hacer esto es escribiendo la directiva `#include` generalmente al principio de un programa.

Estructura de un Programa en C

Es posible utilizar en un programa múltiples funciones, pero siempre debe haber una función principal, de la cual depende el control del programa completo.

En el caso de C se utiliza una función llamada `main()`. Ésta constituye el programa principal y desde ahí se puede hacer uso tanto de las funciones internas como de las externas.

La estructura o sintaxis de un programa creado en C se muestra a continuación en un primer programa elemental.



Estructura de un Programa en C

main()

- Todo programa en C consta de uno o más módulos llamados funciones.
- **Una de las funciones se llama main.** El programa siempre comenzará por la ejecución de la función main, la cual puede acceder a las demás funciones.
- Las definiciones de las funciones adicionales se deben realizar aparte, bien precediendo o siguiendo a main
- Los **paréntesis** identifican a main() como una función. Generalmente, dentro de ellos se incluye información que se envía a la función. En este caso no hay traspaso de información por lo que no hay nada escrito en su interior. Aún así son obligatorios.
- **int** significa que la función retorna un valor.



Estructura de un Programa en C

Cuerpo de una función

- El **cuerpo de una función** (conjunto de sentencias que la componen) va enmarcado entre llaves { }. Ese es el significado de las llaves que aparecen en el ejemplo.
- Las llaves e indican el comienzo de una función, en este caso la función main y su final.
 - El ";" se coloca al final de cada sentencia, es la forma que se usa en **C** para separar una instrucción de otra. Se pueden poner varias en la misma línea siempre que se separen por el punto y coma.



Estructura de un Programa en C

Comentario

`/* Aquí va el cuerpo del programa */`

- El compilador reconoce como comentario cualquier grupo de caracteres situados entre `/*` y `*/`, aunque estén en diferentes líneas.
- Un comentario, no se ejecuta. Sirve para describir el programa. Conviene acostumbrarse a comentar los programas. Un comentario puede ocupar más de una línea
- Se pueden definir comentarios de una sola línea mediante `//`.

`// Este comentario ocupa una sola línea`

- En el caso de comentarios de una sola línea no hay indicador de fin de comentario.



EJEMPLO:

```
#include <stdio.h>
int main ()
{
    printf ("El valor de pi es %f", PI);
    printf ("\n%s", TEXTO);
    return 0;
}
```

PRIMER PROGRAMA EN C