



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): 7

Integrante(s): Ángel Joel Flores Torres

*No. de Equipo de
cómputo empleado:* N/A

No. de Lista o Brigada: 14

Semestre: 1er Semestre

Fecha de entrega: 30 de Noviembre de 2020

Observaciones:

CALIFICACIÓN: _____

OBJETIVO

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

ACTIVIDADES

- Crear un programa en lenguaje C que tenga definidas variables de varios tipos, se les asigne valores adecuados (por lectura o asignación directa) y muestre su valor en la salida estándar.
- En un programa en C, asignar valores a variables utilizando expresiones aritméticas; algunas con uso de cambio de tipo (cast).
- Elaborar expresiones relacionales/lógicas en un programa en C y mostrar el resultado de su evaluación.

INTRODUCCIÓN

Un comentario en programación es la línea de texto en nuestro código fuente que el compilador ignora.

Sabemos que nuestro código fuente está compuesto de instrucciones, y el compilador traduce esas instrucciones del lenguaje de programación que estamos usando a lenguaje máquina.

Cuando el compilador se encuentra con un comentario, esa línea, o varias de ellas, no las traduce, y continúa buscando en la instrucción siguiente.

Los comentarios en programación se utilizan para poner aclaraciones del código, y así es más fácil de entender lo que hace, aunque también se utilizan para anular parte de un código.

Para comentar una o más líneas de código en cualquier vista de editor C/C++. Los caracteres iniciales `//` se añaden al principio de cada línea cuando se comentan una

o varias líneas de código. También puede comentar varias líneas de código en bloque con los caracteres `/* */`.

Declaración de variables

Para declarar variables en C se sigue la siguiente sintaxis:

[modificadores] tipoDeDato identificador [= valor];

Por lo tanto, una variable puede tener modificadores (éstos se analizarán más adelante y son opcionales), debe declarar el tipo de dato que puede contener la variable, debe declarar el identificador (nombre o etiqueta) con el que se va a manejar el valor y se puede asignar un valor inicial a la variable (opcional).

También es posible declarar varios identificadores de un mismo tipo de dato e inicializarlos en el mismo renglón, lo único que se tiene que hacer es separar cada identificador por comas.

tipoDeDato identificador1 [= valor], identificador2 [= valor];

Tipos de datos

Los tipos de datos básicos en C son:

- Caracteres: codificación definida por la máquina.
- Enteros: números sin punto decimal.
- Flotantes: números reales de precisión normal.
- Dobles: números reales de doble precisión.

Para poder acceder al valor de una variable se requiere especificar el tipo de dato.

Los especificadores que tiene lenguaje C para los diferentes tipos de datos son:

<i>Tipo de dato</i>	<i>Especificador de formato</i>
<i>Entero</i>	<code>%d, %i, %ld, %li, %o, %x</code>
<i>Flotante</i>	<code>%f, %lf, %e, %g</code>
<i>Carácter</i>	<code>%c, %d, %i, %o, %x</code>
<i>Cadena de caracteres</i>	<code>%s</code>

El especificador de dato se usa para guardar o imprimir el valor de una variable.

Para imprimir un valor entero en base 10 se pueden usar los especificadores %d o %i (%ld ó %li para enteros largos), para imprimir un valor entero en base 8 se utiliza el especificador %o y para imprimir un valor entero en base 16 se utiliza el especificador %x.

Un valor real se puede imprimir con el especificador %f para reales de precisión simple, %lf para reales de doble precisión, %e (notación científica) y %g (redondea la parte fraccionaria a 3 dígitos significativos).

Una variable de tipo carácter se puede imprimir con el especificador %c, con los especificadores %i o %d para imprimir el valor del código ASCII del carácter en base 10, con el especificador %o para imprimir el valor del código ASCII del carácter en base 8 o con el especificador %x para imprimir el valor del código ASCII del carácter en base 16.

La biblioteca 'stdio.h' contiene diversas funciones tanto para imprimir en la salida estándar (monitor) como para leer de la entrada estándar (teclado).

printf es una función para imprimir con formato, es decir, se tiene que especificar entre comillas el tipo de dato que se desea imprimir, también se puede combinar la impresión de un texto predeterminado.

Mientras que scanf es una función que sirve para leer datos de la entrada estándar (teclado), para ello únicamente se especifica el tipo de dato que se desea leer entre comillas y en qué variable se quiere almacenar. Al nombre de la variable le antecede un ampersand (&), esto indica que el dato recibido se guardará en la localidad de memoria asignada a esa variable.

Para imprimir con formato también se utilizan algunas secuencias de caracteres de escape, C maneja los siguientes:

\a carácter de alarma

\b retroceso

\f avance de hoja

\n salto de línea

\r regreso de carro

\t tabulador horizontal

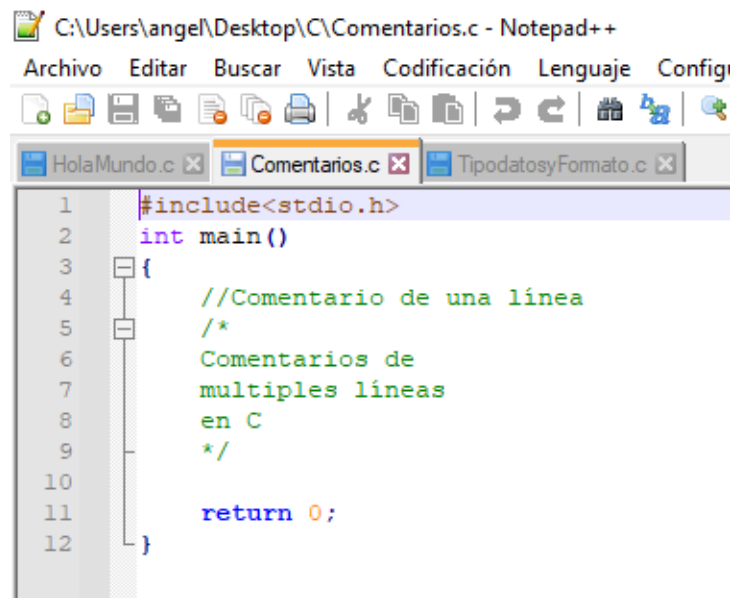
\v tabulador vertical

'\0' carácter nulo

RESULTADOS

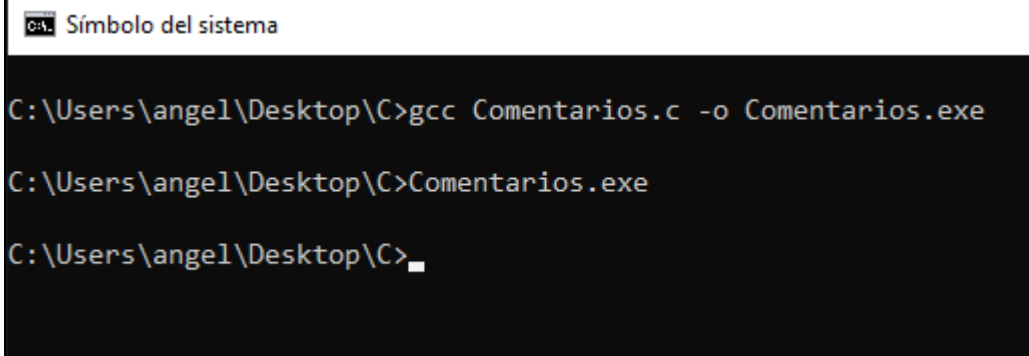
Comentarios

Aquí aprendí que existen comentario de una línea y de varias que sirven para hacer más fácil el entendimiento de cada línea de código



The screenshot shows a Notepad++ window titled "C:\Users\angel\Desktop\C\Comentarios.c - Notepad++". The menu bar includes "Archivo", "Editar", "Buscar", "Vista", "Codificación", "Lenguaje", and "Configuración". The toolbar contains various icons for file operations and editing. The editor has three tabs: "HolaMundo.c", "Comentarios.c" (active), and "TipodatosyFormato.c". The code in "Comentarios.c" is as follows:

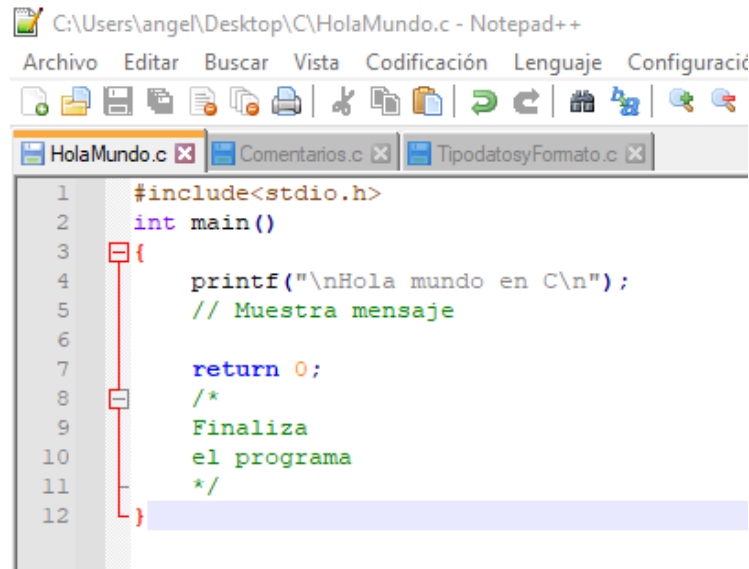
```
1  #include<stdio.h>
2  int main()
3  {
4      //Comentario de una línea
5      /*
6      Comentarios de
7      multiples líneas
8      en C
9      */
10
11     return 0;
12 }
```



The screenshot shows a Windows command prompt window titled "C:\> Símbolo del sistema". The commands entered and their outputs are:

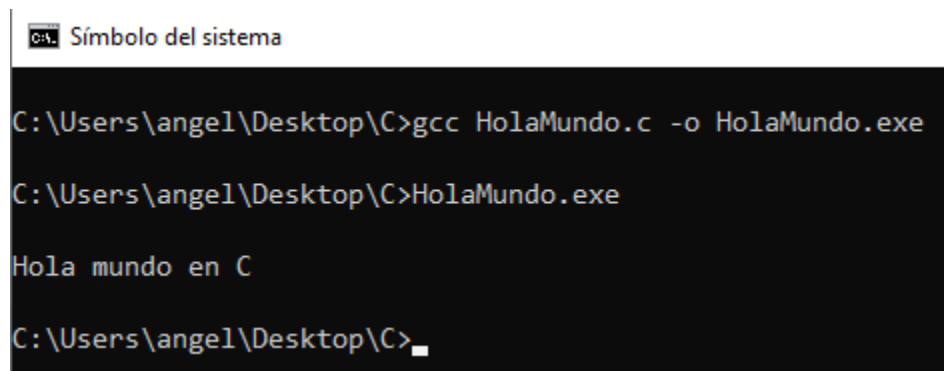
```
C:\Users\angel\Desktop\C>gcc Comentarios.c -o Comentarios.exe
C:\Users\angel\Desktop\C>Comentarios.exe
C:\Users\angel\Desktop\C>.
```

Y lo aplique en mi hola mundo. Cabe mencionar que los comentarios no salen a la hora de ejecutar el programa, simplemente son para facilitar al usuario el entendimiento del mismo.



The screenshot shows the Notepad++ editor with the file 'C:\Users\angel\Desktop\C\HolaMundo.c' open. The code is as follows:

```
1  #include<stdio.h>
2  int main()
3  {
4      printf("\nHola mundo en C\n");
5      // Muestra mensaje
6
7      return 0;
8      /*
9      Finaliza
10     el programa
11     */
12 }
```



The screenshot shows a Windows command prompt window titled 'Símbolo del sistema'. The commands and output are as follows:


```
C:\Users\angel\Desktop\C>gcc HolaMundo.c -o HolaMundo.exe
C:\Users\angel\Desktop\C>HolaMundo.exe
Hola mundo en C
C:\Users\angel\Desktop\C>
```

Tipo de datos y formatos

Carácter

Lo más destacable de este apartado es que para utilizar caracteres especiales como los acentos, etc, se utiliza la tabla del código ASCII, y se declara la variable junto con el valor, y así se pondrá el carácter.

```
//C  acteres
char car='m';
char acentoa=160;
char acentou=163;
char acentoo=162;
printf("C  cracter: %c \n",acentoa,car);
printf("C  cracter en decimal: %d \n",acentoa,car);
printf("C  cracter en decimal: %i \n",acentoa,car);
printf("C  cracter en octal: %o \n",acentoa,car);
printf("C  cracter en hexadecimal: %x \n\n",acentoa,car);
```

 S  mbolo del sistema

```
C:\Users\angel\Desktop\C>gcc DatosyFormato.c -o DatosyFormato.exe
C:\Users\angel\Desktop\C>DatosyFormato.exe

Caracter: m

Caracter en decimal: 109

Caracter en decimal: 109

Caracter en octal: 155

Caracter en hexadecimal: 6d
```

N  meros enteros cortos

```
//N  meros enteros cortos
short enteroc=17120;
printf("N  cmmero entero corto: %i \n",acentou,enteroc);
printf("N  cmmero entero corto: %d \n",acentou,enteroc);
printf("N  cmmero entero corto en octal: %o \n",acentou,enteroc);
printf("N  cmmero entero corto en hexadecimal: %x \n\n",acentou,enteroc);
int enteroc2=-1222;
printf("N  cmmero entero corto: %i \n",acentou,enteroc2);
printf("N  cmmero entero corto: %d \n",acentou,enteroc2);
printf("N  cmmero entero corto en octal: %o \n",acentou,enteroc2);
printf("N  cmmero entero corto en hexadecimal: %x \n\n",acentou,enteroc2);
```

```

Número entero corto: 17120
Número entero corto: 17120
Número entero corto en octal: 41340
Número entero corto en hexadecimal: 42e0

Número entero corto: -1222
Número entero corto: -1222
Número entero corto en octal: 3777775472
Número entero corto en hexadecimal: fffffb3a

```

Números enteros largos

```

//Números enteros largos
long enterol=17777777;
printf("Número entero largo: %ld \n",acentou,enterol);
printf("Número entero largo: %li \n\n",acentou,enterol);

```

```

Número entero largo: 17777777
Número entero largo: 17777777

```

Números reales cortos

```

//Números reales cortos
float realc=12.171520;
printf("Número real corto: %f \n",acentou,realc);
printf("Número real corto redondeado: %g \n",acentou,realc);
printf("Número real corto en notación científica: %e \n\n",acentou,acentoo,realc);

```

```

Número real corto: 12.171520
Número real corto redondeado: 12.1715
Número real corto en notación científica: 1.217152e+001

```

Números reales largos

```

//Números reales largos
double reall=12.15132135213213;
printf("Número real largo: %lf \n",acentou,reall);
printf("Número real largo: %.14lf \n",acentou,reall);

```



```
Número real largo: 12.151321
Número real largo: 12.15132135213213
```

Sentencias de escape

```
char acentoi=161;
//Salto de línea
printf("---Salto de línea---\n",acentoi);
printf("Hola mundo\n");
printf("Bienvenidos al curso de C\n\n");

//Tabulador horizontal
printf("---Tabulador horizontal---\n");
printf("Hola mundo\t");
printf("Bienvenidos al curso de C\n\n");

//Caracter de alarma
printf("---Caracter de alarma---\n");
printf("Hola mundo\a");
printf("Bienvenidos al curso de C\n\n");

//Retroceso de carro
printf("---Retroceso de carro---\n");
printf("Hola mundo\r");
printf("Bienvenidos al curso de C\n\n");

//Retroceso
printf("---Retroceso---\n");
printf("Hola mundo\b");
printf("Bienvenidos al curso de C\n\n");
```

```
C:\Users\angel\Desktop\C>SentenciasEscape.exe
---Salto de línea---
Hola mundo
Bienvenidos al curso de C

---Tabulador horizontal---
Hola mundo      Bienvenidos al curso de C

---Caracter de alarma---
Hola mundo Bienvenidos al curso de C

---Retroceso de carro---
Bienvenidos al curso de C

---Retroceso---
Hola mund Bienvenidos al curso de C
```

CONCLUSIÓN

La práctica me resulto bastante interesante, y más por el hecho de que nunca había programado mediante un editor de texto y una terminal, durante mi carrera técnica de la preparatoria estuve acostumbrado a utilizar un IDE en el que se escribía el código y así mismo se compilaba.

Acostumbrarme no me ha costado tanto trabajo gracias a que he estado familiarizado con estos términos, como los tipos de datos, las variables, las sentencias de escape, etc.

Considero que esta práctica es muy importante pues lo fundamental de programación en C se centra más en esta práctica, sin despreciar a las anteriores, y su importancia radica en el hecho que el profesor había mencionado con anterioridad que a la hora de buscar empleo la mayoría de las entrevistas te piden programar un hola mundo, y solo te proporcionan un editor de texto y una terminal previamente configurada.

REFERENCIAS

¿Qué es un COMENTARIO en programación? (s. f.). Lenguajes de programación.

Recuperado 29 de noviembre de 2020, de

<https://lenguajesdeprogramacion.net/diccionario/que-es-un-comentario-en-programacion/>

IBM Knowledge Center. (s. f.). *Comentar código*. IBM. Recuperado 29 de noviembre

de 2020, de

https://www.ibm.com/support/knowledgecenter/es/SSQ2R2_9.1.1/org.eclipse.cdt.doc.user/tasks/cdt_t_comment_out.htm

Microsoft. (2018). *Comentarios en C*. Microsoft Docs. <https://docs.microsoft.com/es-es/cpp/c-language/c-comments?view=msvc-160#:~:text=Un%20%E2%80%9Ccomentario%E2%80%9Des%20una%20secuencia,alto%20de%20cualquier%20otra%20manera.>