	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de Programacion

Grupo: 3

No de Práctica(s): 7

Integrante(s): Meraz Dionicio Israel

No. de Equipo de cómputo empleado: 2

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 3 de Octubre del 2019

Observaciones: Muy bien

CALIFICACIÓN: **10**

OBJETIVO

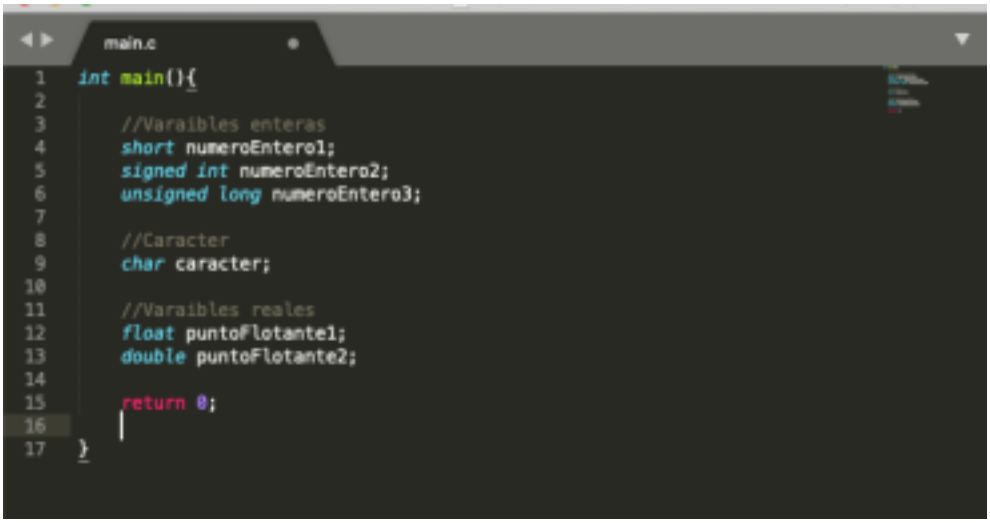
Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo de secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

INTRODUCCION

El lenguaje de programación C fue creado por Brian Kernighan y Dennis Ritchie a mediados de los años 70. La primera implementación del mismo la realizó Dennis Ritchie sobre un computador DEC PDP-11 con sistema operativo UNIX.

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

- Tipos de variables



Una variable es donde se guardan los datos que se ingresa en un programa, pero primero hay que declararla que tipo de variable es:

Para los enteros se representan en int
Para los reales en float o double
Para un carácter es char

DATA TYPE	MEMORY (BYTES)	RANGE
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	-(2^63) to (2^63)-1
unsigned long long int	8	0 to 18,446,744,073,709,551,615

Existen variaciones en el int y double como se muestran las siguientes tablas, esto se debe a la cantidad de bits que utilizara la computadora para almacenar los valores o caracteres que ingreses en las variables.

Tipo	*Bits*	*Valor Mínimo*	*Valor Máximo*
float	32	3.4 E-38	3.4 E38
double	64	1.7 E-308	1.7 E308
long double	80	3.4 E-4932	3.4 E4932

Ubicándonos en la terminal que compila y se corre

```
Last login: Mon Sep 30 09:00:47 on console
Laos03:~ fp03alu29$ pwd
/Users/fp03alu29
Laos03:~ fp03alu29$ cd Documents/
Laos03:Documents fp03alu29$ main.c
-bash: main.c: command not found
Laos03:Documents fp03alu29$ gcc main.c -o main
Laos03:Documents fp03alu29$ ./main
Laos03:Documents fp03alu29$
```

Al correr el programa no hace nada porque solo estas declarando variables

- Mostrar y leer

```c
int main(){

    //Declaramos variables a leer
    int numeroEntrada;
    double realEntrada;

    //Asignamos Varables
    int numeroEntero = 3768;
    char caracter = 'B';
    float numeroReal = 89.8;

    //Mostramos texto y valores
    printf("Primero texto solo\n");
    printf("Luego podemos poner un entero: %i\n", numeroEntero);
    printf("Tambien pidemos poner un caracter: %c\n", caracter);
    printf("Y un numero real: %.2f\n", numeroReal);

    //Leemos valores
    scanf("%i", &numeroEntrada);
    scanf("%lf", &realEntrada);

    //Y nahora podemos mostrarlos tambien
    printf("Tu entero: %i\n", numeroEntrada);
    printf("Tu real: %.2lf\n", realEntrada);

    return 0;

}
```

Lo primero es declarar las variables que vamos a ocupar.
Los // representan a comentarios, el programa lo va a ignorar.
Se declaró los valores que están definidas, igualmente se declararon las variables, nota si es un carácter se utiliza las ‘’

Para mostrar valores es con printf, se abre paréntesis y se abren comillas “” para especificar si es un entero u otro formato se utilizan los símbolos que están en la tabla de abajo. Después de las comillas la variable que vas a mostrar
Para leer es scanf entre comillas la especificación después la variable

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

Compilando y corriendo

```
Laos03:Documents fp03alu29$ pwd
/Users/fp03alu29/Documents
Laos03:Documents fp03alu29$ gcc int.c -o int
Laos03:Documents fp03alu29$ ./int
Primero texto solo
Luego podemos poner un entero: 3768
Tambien pidemos poner un caracter: B
Y un numero real: 89.80
78
87.6
Tu entero: 78
Tu real: 87.60
Laos03:Documents fp03alu29$
```

- Operadores

```c
#include <stdio.h>

int main() {

    int dos, tres, cuatro, cinco;
    double resultado;

    dos=2;
    tres=3;
    cuatro=4;
    cinco=5;

    resultado = cinco/dos;
    printf("5 / 2 = %llf\n", resultado);

    resultado = (double)cinco/dos;
    printf("5 / 2 = %llf\n", resultado);

    return 0;

}
```

La forma de utilizar y escribir operadores en un programa es la siguiente:
Para la suma es con +
Para la resta es con –
La multiplicación es con *
La división es con /
Y el modulo es con %

Se observa que el resultado = cinco/dos; se repitió pero la diferencia es que el primero el resultado te dará un entero porque divide entre enteros
Y el segundo se declaró que el resultado lo quieres en doublé

NOTA: en %.1lf\n, el .1 representa la cantidad de dígitos que va a utilizar después del punto.
El #include <stdio.h> es una libreria la que define funciones.

Compilando y corriendo el programa

```
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ gcc operador.c -o operador

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./operador
5 / 2 = 2.000000
5 / 2 = 2.500000

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$
```

- Operadores lógicos y comparaciones

```c
#include <stdio.h>

int main() {

    int num1, num2, res;
    char c1, c2;

    num1 = 7;
    num2 = 15;
    c1 = 'h';
    c2 = 'H';

    printf(" ¿ num1 es menor a num2 ? -> \t%d\n", num1<num2);
    printf("¿ c1 es igual a c2 ? -> \t%d\n", c1==c2);
    printf("¿ c1 es diferente a c2 -> \t%d\n", c1!=c2);

    res = num1 < num2 && c1 == 'h';
    printf("¿ num1 < num2 Y c1 es igual a 'h' ? -> \t%d\n", res);

    res = c1 == 'a' || c2 == 'H';
    printf("¿c1es igual a 'a' O c2 a 'H' -> \t%d\n", res);

    return 0;

}
```

Los operadores lógicos se utilizan para que nos proporcione si es cierto o no una condición.

El cero representa que es falso y el uno como verdadero
Si quieres comparar o utilizar los operadores lógicos se tienen que escribir después de las comillas peron aun dentro del printf

Operador	*Operación*	*Uso*	*Resultado*
==	Igual que	'h' == 'H'	Falso
!=	Diferente a	'a' != 'b'	Verdadero
<	Menor que	7 < 15	Verdadero
>	Mayor que	11 > 22	Falso
<=	Menor o igual	15 <= 22	Verdadero
>=	Mayor o igual	20 >= 35	Falso

Compilando y corriendo

```
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ gcc logico.c -o logico

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./logico
 ¿ num1 es menor a num2 ? ->    1
¿ c1 es igual a c2 ? ->         0
¿ c1 es diferente a c2 ->       1
¿ num1 < num2 Y c1 es igual a 'h' ? ->  1
¿c1es igual a 's' O c2 a 'H' ->         1

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$
```

Esta tabla tiene el operador para comparar variables ya sea en números o caracteres

La operación NO se le tiene que aplicar su viceversa por ejemplo si es verdadero el resultado sería falso
El AND, el resultado será verdadero si y si solo si las dos variables sean verdaderos, c1== 'a' se consideran como una variable
El OR, si una variable es verdadera, el resultado será verdadero

Operador	*Operación*
!	No
&&	Y
||	O

CONCLUSIONES

Utilizar el lenguaje C no es sencillo de dominarlo en poco tiempo ya que tenemos que analizar y tener ordenado nuestros procesos para que la computadora la pueda interpretar, apenas estamos conociendo este nuevo leguaje.

Los fundamentos de la programación contiene la forma de escribir una variable y declararla, como mostrar y/o leer valores en la computadora, la utilización de los operadores para que el programa las pueda realizar, las comparaciones y operadores lógicos que muestran resultados como verdadero o falso.