


|  |                        |                         |
|--|------------------------|-------------------------|
|  | <b>Práctica</b>        |                         |
|  | Facultad de Ingeniería | Laboratorio de docencia |

# Laboratorios de computación salas A y B

|   |                             |
|---|-----------------------------|
| <i>Profesor:</i>                              | Alejandro Pimentel          |
| <i>Asignatura:</i>                            | Fundamentos de Programación |
| <i>Grupo:</i>                                 | Grupo 3 Bloque 135          |
| <i>No de Práctica(s):</i>                     | Practica #7                 |
| <i>Integrante(s):</i>                         | Muñoz Reyes Laura Vanessa   |
| <i>No. de Equipo de<br/>cómputo empleado:</i> | Luxemburgo 10               |
| <i>No. de Lista o Brigada:</i>                | No. de Cuenta 3177522823    |
| <i>Semestre:</i>                              | 2020-01                     |
| <i>Fecha de entrega:</i>                      | 03 de octubre del 2019      |
| <i>Observaciones:</i>                         |                             |

**CALIFICACIÓN:** \_\_\_\_\_

## **Practica #7.**

### **Fundamentos del Lenguaje C**

#### **INTRODUCCION:**

Una vez que tenemos un problema vemos nuestras variables y a el resultado que queremos llegar, como ya hemos visto hay distintas maneras de expresar nuestro problema ya sea en algoritmo o en diagrama de flujo sin embargo es importante ya codificar y pasarlo para que nuestra computadora pueda ejecutarlo. Para realizar esto es importante escoger un buen lenguaje y tener los fundamentos necesarios para aplicarlo.

#### **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.

#### **MARCO REFERENCIAL:**

Una vez que un problema dado ha sido analizado (se identifican los datos de entrada y la salida deseada), que se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), y que se ha representado el algoritmo de manera gráfica o escrita (diagrama de flujo o pseudocódigo) se puede proceder a la etapa de codificación. La codificación se puede realizar en cualquier lenguaje de programación estructurada, como lo son Pascal, Python, o C.

La programación en C se utiliza, entre otras cosas, para el desarrollo de sistemas operativos. Uno de los primeros sistemas operativos escrito en C fue Unix. Más tarde, Linux. Pero C no es sólo el lenguaje de programación de los sistemas operativos. C es el precursor e inspirador para casi todos los lenguajes de alto nivel más populares disponibles en la actualidad. De hecho, Perl, PHP, Python y Ruby están escritos mediante programación en C.

Presentando así muchas ventajas el aprender programación en C antes de otros, gracias a que en C tienes un par de llamadas a librerías, después tienes que mantener la concentración en la semántica del lenguaje, y a menudo te obliga a pensar más y más profundo acerca de lo que está realmente sucediendo.

#### **Tipos de variables**

El lenguaje C obliga a declarar una variable antes de ser usada, declarar una variable no significa que se le asigne contenido, sino simplemente se indica que la variable existe. En cuanto a los tipos de variables más comunes son:

| TIPO DE DATOS  | SE ESCRIBE | MEMORIA REQUERIDA* | RANGO ORIENTATIVO*                              | EQUIVALENCIA EN PSEUDOCÓDIGO | OBSERVACIONES                                 |
|----------------|------------|--------------------|---|------------------------------|---|
| Entero         | int        | 2 bytes            | - 32768 a 32767                                 | Entero                       | Uso en contadores, control de bucles etc.     |
| Entero largo   | long       | 4 bytes            | - 2147483648 a 2147483647                       | Entero                       | Igual que int pero admite un rango más amplio |
| Decimal simple | float      | 4 bytes            | - $3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$     | Real                         | Hasta 6 decimales. También admite enteros     |
| Decimal doble  | double     | 8 bytes            | - $1,79 \cdot 10^{308}$ a $1,79 \cdot 10^{308}$ | Real                         | Hasta 14 decimales. También admite enteros    |
| Carácter       | char       | 1 bytes            | 0 a 255   | Alfanumérica                 | Carácter, independiente o parte de una cadena |

Para los reales, se tienen también diferentes tipos de variables que asignan más bits para tener mayor rango y mayor precisión. Las variables reales siempre poseen signo.

| Tipo          | Descripción              | Bits | Rango  |
|---------------|--------------------------|------|--|
| unsigned char | carácter sin signo       | 8    | 0 a 255  |
| char          | carácter                 | 8    | -128 a 127                                       |
| short int     | entero corto             | 16   | -32768 a 32767                                   |
| unsigned int  | entero sin signo         | 32   | 0 a 4 294 967 295                                |
| int           | entero                   | 32   | -2 147 483 648 a 2 147 483 647                   |
| unsigned long | entero largo sin signo   | 32   | 0 a 4 294 967 295                                |
| enum          | enum                     | 16   | -2 147 483 648 a 2 147 483 647                   |
| long          | entero largo             | 32   | -2 147 483 648 a 2 147 483 647                   |
| float         | real (con punto decimal) | 32   | $3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$     |
| double        | real doble               | 64   | $1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$   |
| long double   | real doble largo         | 80   | $3,4 \times 10^{-4932}$ a $1,1 \times 10^{4932}$ |

## Mostrar y leer

Otra tarea que debe enfrentar nuestro programa es la de pedir datos al usuario, por ejemplo que introduzca un número o una palabra. Esto se realiza a través del comando *scanf*

El indicador de tipo de dato hace explícito el tipo de dato que se espera recibir en la entrada. Se usan los mismos códigos para mostrar datos en pantalla con *printf*. Los indicadores de tipos principales son los siguientes:

| Código de formato | Significado                           |
|-------------------|---------------------------------------|
| %d                | Tipo entero (int)                     |
| %ld               | Tipo entero largo (long)              |
| %lf               | Tipo decimal doble precisión (double) |

|    |  |
|----|--|
| %c | Tipo carácter                                  |
| %s | Tipo cadena alfanumérica (array de caracteres) |

Existen más códigos de formato (por ejemplo %e, %f ó %g para tipo float) y es posible introducir modificadores para especificar precisión decimal o longitud de los datos, pero con estos códigos básicos es suficiente para las construcciones más habituales en programas.

## Operadores

En C hay muchos operadores de asignación que serán en realidad simplificaciones de expresiones más complejas, como las auto-operaciones (+=, -=, \*=, /=, &=,...) y los pre/post incrementos/decrementos (++ , --).

| DdF        | C | Descripción              |
|------------|---|--------------------------|
| +          | + | Suma, adición            |
| -          | - | Resta, sustracción       |
| .          | * | Multiplicación, producto |
| <u>Div</u> | / | Cociente división entera |
| <u>Mod</u> | % | Resto división entera    |
| /          | / | División                 |

## Comparaciones

| DdF | C  | Descripción   |
|-----|----|---------------|
| >   | >  | Mayor         |
| ≥   | >= | Mayor o igual |
| <   | <  | Menor         |
| ≤   | <= | Menor o igual |
| =   | == | Igual         |
| ≠   | != | Diferente     |

## Operadores Lógicos

Los operandos son considerados falsos (valor 0) ó ciertos (valor distinto de 0). El resultado siempre es 0 ó 1.

| DdF | C  | Descripción        |
|-----|----|--------------------|
| ^   | && | And, y, conjunción |
| v   |    | Or, o, disyunción  |
| ¬   | !  | Not, no, negación  |

## PROCEDIMIENTO / RESULTADO:

### 1. Realiza los siguientes programas en Sublime Text, compilar y correr.

#### 1.1 Tipos de Variables

**Nota:** Se realizaron los programas en Sublime Text y al momento de compilar y correr como se realizó en Windows se usó un programa alterno MinGW y también se investigó que para compilar en Windows el archivo de salida debe tener terminación .exe y esta misma terminación es la que utilizamos al correr o ejecutar. Este procedimiento puede visualizarse en las capturas de la terminal.

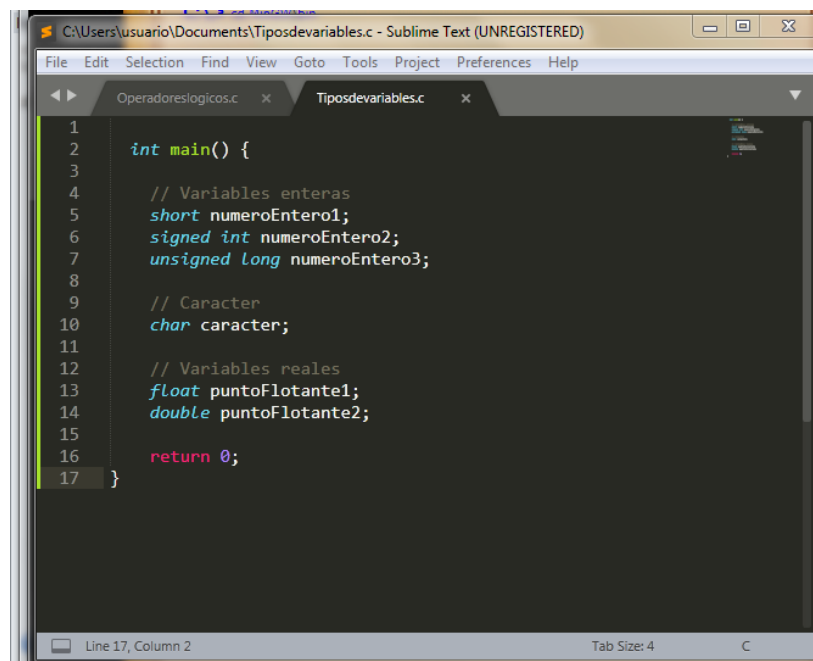
#### Programa dado

##### Tipos de variables

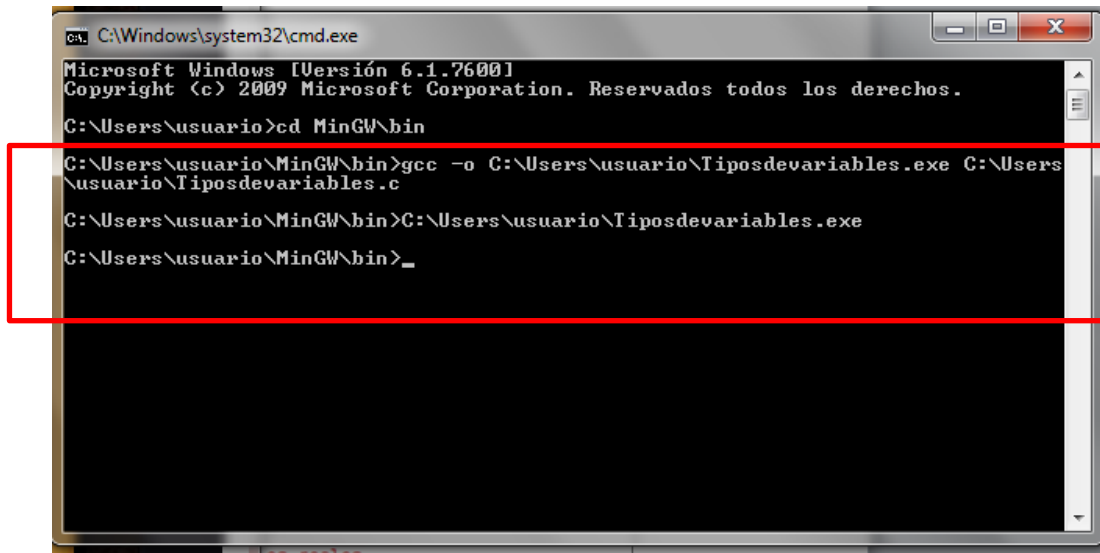
```
int main() {  
  
    // Variables enteras  
    short numeroEntero1;  
    signed int numeroEntero2;  
    unsigned long numeroEntero3;  
  
    // Caracter  
    char caracter;  
  
    // Variables reales  
    float puntoFlotante1;  
    double puntoFlotante2;  
  
    return 0;  
}
```

Página 5 de 12

#### Programa en Lenguaje C



Se compilo y se corrió en la terminal



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\usuario>cd MinGW\bin

C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Tiposdevariables.exe C:\Users\usuario\Tiposdevariables.c

C:\Users\usuario\MinGW\bin>C:\Users\usuario\Tiposdevariables.exe

C:\Users\usuario\MinGW\bin>_
```

## 1.2 Mostrar y leer

Programa dado



```
Mostrar y Leer

#include <stdio.h>

int main() {

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

    // Asignamos variables
    int numeroEntero = 32768;
    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("También podemos poner un caracter: %c\n", caracter);
    printf("Y un numero real: %.2f\n", numeroReal);

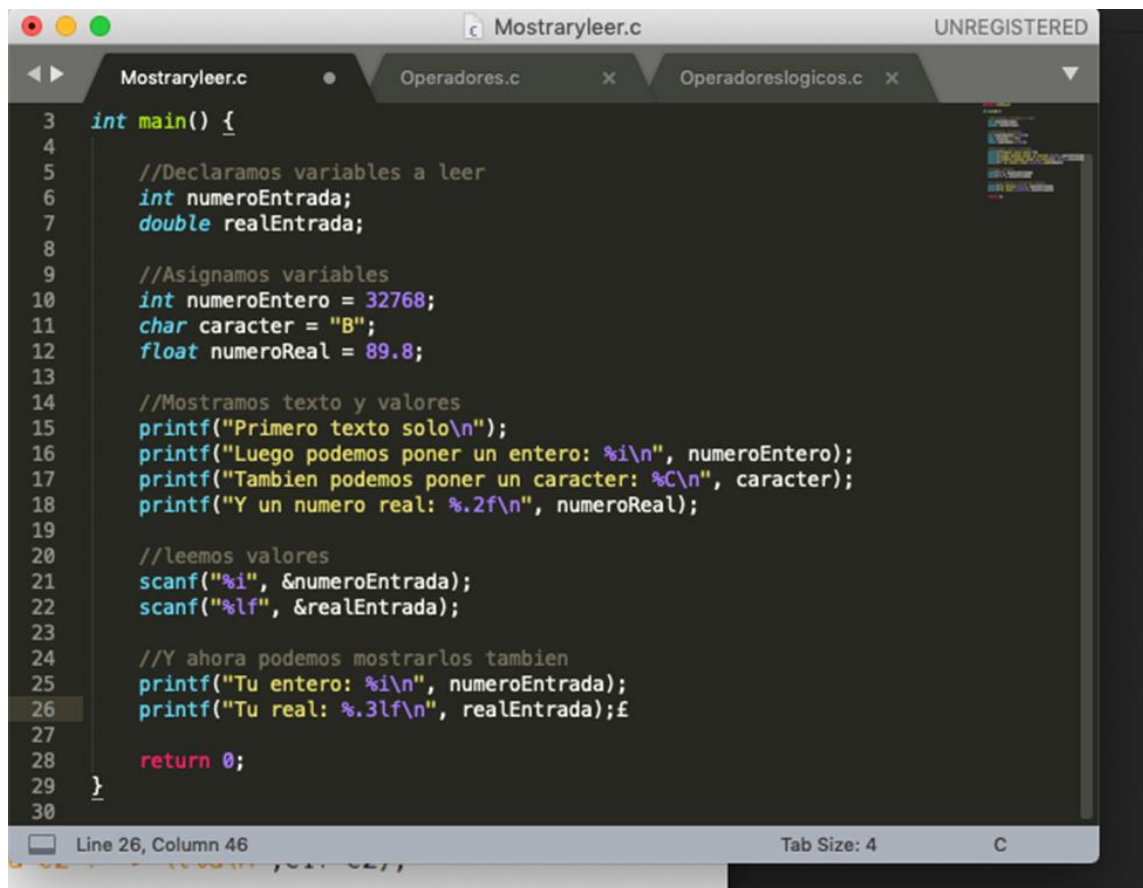
    // Leemos valores
    scanf("%i", &numeroEntrada);
    scanf("%lf", &realEntrada);

    // Y ahora podemos mostrarlos también
    printf("Tu entero: %i\n", numeroEntrada);
    printf("Tu real: %.3lf\n", realEntrada);

    return 0;
}
```

Página 7 de 12 7 / 13

## Programa en lenguaje c

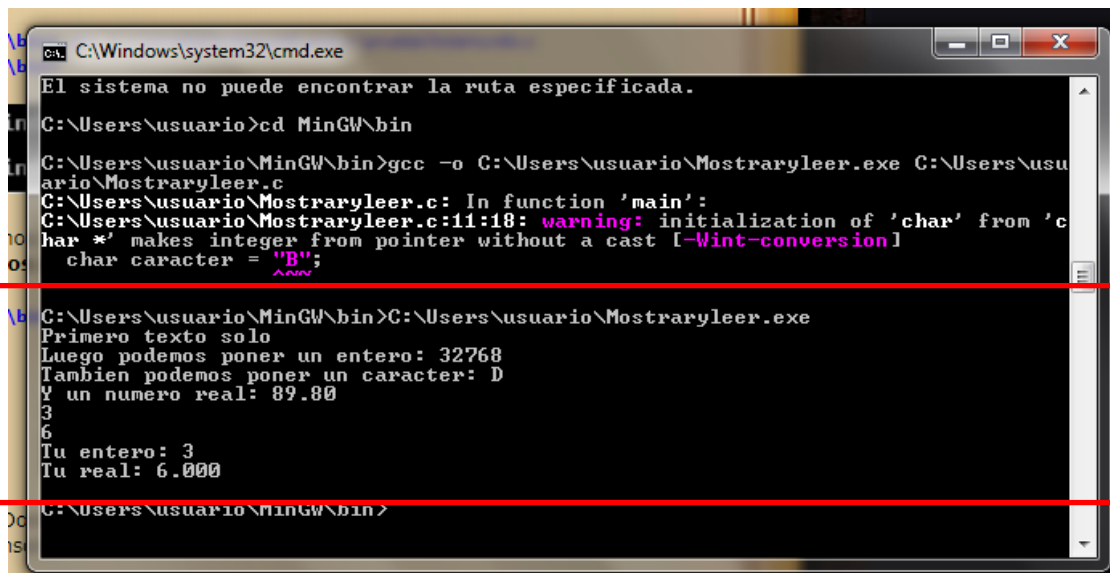


The screenshot shows a code editor window titled 'Mostraryleer.c' with a 'UNREGISTERED' watermark. The code is as follows:

```
3 int main() {
4
5     //Declaramos variables a leer
6     int numeroEntrada;
7     double realEntrada;
8
9     //Asignamos variables
10    int numeroEntero = 32768;
11    char caracter = "B";
12    float numeroReal = 89.8;
13
14    //Mostramos texto y valores
15    printf("Primero texto solo\n");
16    printf("Luego podemos poner un entero: %i\n", numeroEntero);
17    printf("Tambien podemos poner un caracter: %C\n", caracter);
18    printf("Y un numero real: %.2f\n", numeroReal);
19
20    //leemos valores
21    scanf("%i", &numeroEntrada);
22    scanf("%lf", &realEntrada);
23
24    //Y ahora podemos mostrarlos tambien
25    printf("Tu entero: %i\n", numeroEntrada);
26    printf("Tu real: %.3lf\n", realEntrada);
27
28    return 0;
29 }
30
```

The status bar at the bottom indicates 'Line 26, Column 46', 'Tab Size: 4', and 'C'.

Se compilo y se corrió en la terminal. Al compilar salió warning más no error por lo que al momento de correr la terminal genero bien lo que se esperaba



The screenshot shows a Windows command prompt window with the following commands and output:

```
C:\Windows\system32\cmd.exe
El sistema no puede encontrar la ruta especificada.
C:\Users\usuario>cd MinGW\bin
C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Mostraryleer.exe C:\Users\usuario\Mostraryleer.c
C:\Users\usuario\MinGW\bin>C:\Users\usuario\Mostraryleer.exe
Primero texto solo
Luego podemos poner un entero: 32768
Tambien podemos poner un caracter: D
Y un numero real: 89.80
3
6
Tu entero: 3
Tu real: 6.000
C:\Users\usuario\MinGW\bin>
```

A red rectangle highlights the execution output, which matches the expected program behavior.

## 1.3 Operadores

### Programa dado

#### Operadores

```
#include <stdio.h>

int main() {

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

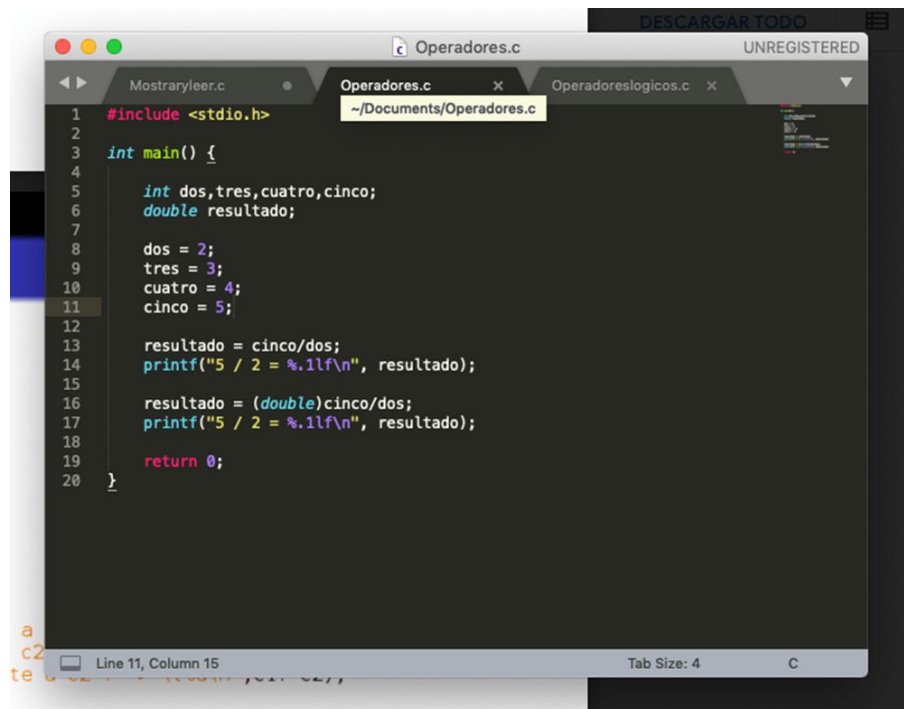
    dos = 2;
    tres = 3;
    cuatro = 4;
    cinco = 5;

    resultado = cinco/dos;
    printf("5 / 2 = %.1lf\n", resultado);

    resultado = (double)cinco/dos;
    printf("5 / 2 = %.1lf\n", resultado);

    return 0;
}
```

### Programa en lenguaje C





Se compilo y se corrió en la terminal

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\usuario>dir
El volumen de la unidad C es WINDOWS
El número de serie del volumen es: 3E39-4654

Directorio de C:\Users\usuario

03/10/2019  09:29 p.m.  <DIR>          .
03/10/2019  09:29 p.m.  <DIR>          ..
02/05/2018  10:13 p.m.  <DIR>          .fontconfig
25/08/2019  01:15 p.m.          106 .gitconfig
26/08/2019  01:26 a.m.  <DIR>          .VirtualBox
13/08/2013  02:34 p.m.          0 agent.log
13/08/2013  01:05 p.m.  <DIR>          Contacts
03/10/2019  08:47 p.m.  <DIR>          Desktop
03/10/2019  12:44 a.m.  <DIR>          Documents
03/10/2019  09:16 p.m.  <DIR>          Downloads
06/08/2019  11:24 p.m.  <DIR>          Dropbox
13/08/2013  01:05 p.m.  <DIR>          Favorites
25/08/2019  11:54 p.m.  <DIR>          FP_2020-1_2823
13/08/2018  11:22 p.m.  <DIR>          Links
03/10/2019  09:17 p.m.  <DIR>          MinGW
30/09/2019  10:08 a.m.          658 Mostraryleer.c
03/10/2019  09:29 p.m.    42.838 Mostraryleer.exe
13/08/2013  01:05 p.m.  <DIR>          Music
17/03/2015  04:08 p.m.  <DIR>          OneDrive
30/09/2019  10:12 a.m.          278 Operadores.c
30/09/2019  10:24 a.m.          482 Operadoreslogicos.c
02/10/2019  12:55 a.m.  <DIR>          Pictures
26/08/2019  12:40 a.m.  <DIR>          PROYECTO
01/01/2014  06:42 p.m.  <DIR>          Saved Games
04/01/2014  03:56 p.m.  <DIR>          Searches
16/06/2015  05:17 p.m.  <DIR>          Tracing
06/07/2019  11:37 p.m.  <DIR>          Videos
19/08/2019  11:26 p.m.  <DIR>          VirtualBox UMs
        6 archivos          44.362 bytes
       22 dirs    1.038.712.832 bytes libres

C:\Users\usuario>cd MinGW\bin

C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Operadores.exe C:\Users\usuario\Operadores.c

C:\Users\usuario\MinGW\bin>C:\Users\usuario\Operadores.exe
5 / 2 = 2.0
5 / 2 = 2.5

C:\Users\usuario\MinGW\bin>
```

## 1.4 Operadores Lógicos

Programa dado

### Operadores lógicos

```
#include <stdio.h>

int main() {

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

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

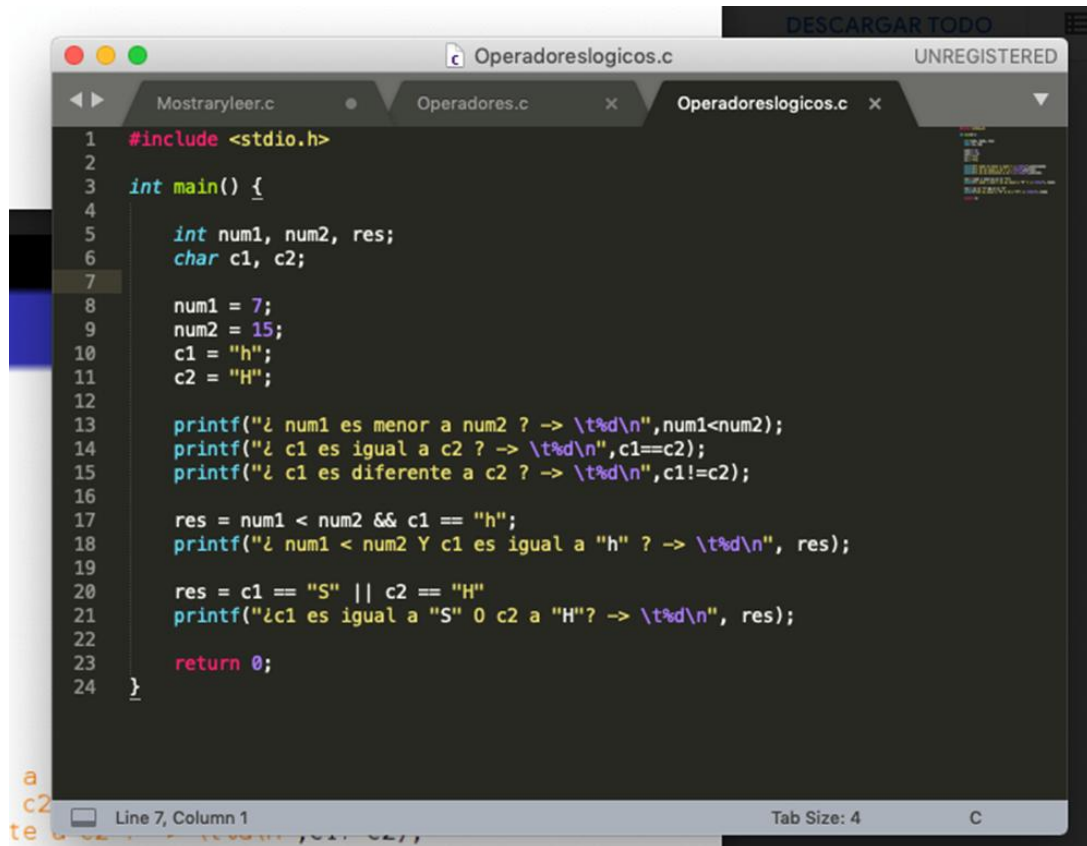
    printf("%d num1 es menor a num2 ? -> %d\n", num1 < num2);
    printf("%d c1 es igual a c2 ? -> %d\n", c1 == c2);
    printf("%d c1 es diferente a c2 ? -> %d\n", c1 != c2);

    res = num1 < num2 && c1 == 'h';
    printf("%d num1 < num2 Y c1 es igual a 'h' ? -> %d\n", res);

    res = c1 == 's' || c2 == 'H';
    printf("%d c1 es igual a 's' O c2 a 'H' ? -> %d\n", res);

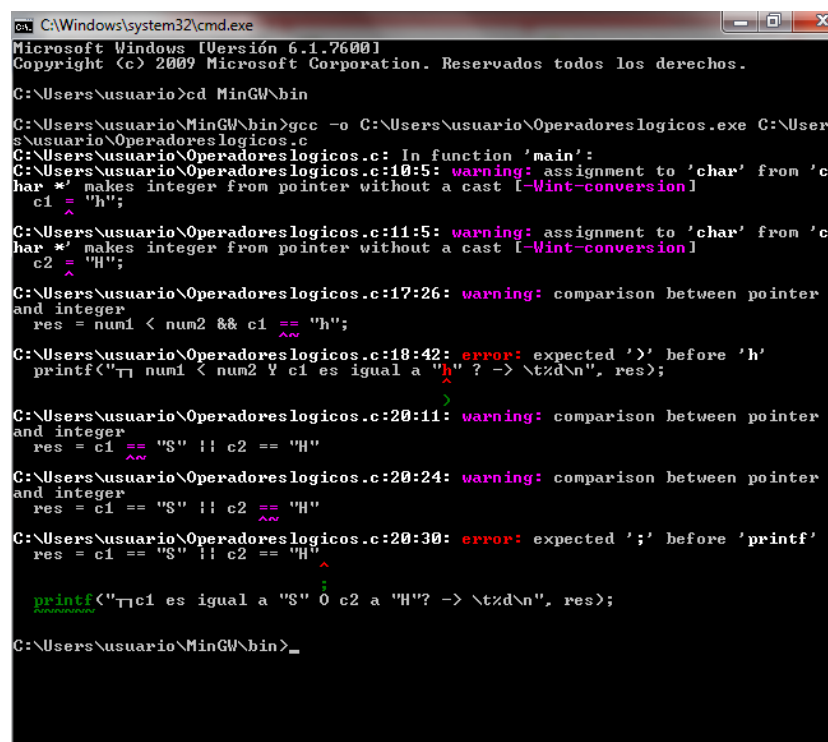
    return 0;
}
```

## Programa en lenguaje c



```
1 #include <stdio.h>
2
3 int main() {
4
5     int num1, num2, res;
6     char c1, c2;
7
8     num1 = 7;
9     num2 = 15;
10    c1 = "h";
11    c2 = "H";
12
13    printf("\n num1 es menor a num2 ? -> \t%d\n", num1 < num2);
14    printf("\n c1 es igual a c2 ? -> \t%d\n", c1 == c2);
15    printf("\n c1 es diferente a c2 ? -> \t%d\n", c1 != c2);
16
17    res = num1 < num2 && c1 == "h";
18    printf("\n num1 < num2 Y c1 es igual a 'h' ? -> \t%d\n", res);
19
20    res = c1 == "S" || c2 == "H";
21    printf("\n c1 es igual a 'S' O c2 a 'H'? -> \t%d\n", res);
22
23    return 0;
24 }
```

Al momento de compilar el programa se obtuvo error por lo que volvimos a revisar el programa en sublime text y se le realizaron las modificaciones necesarias.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\usuario>cd MinGW\bin

C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Operadoreslogicos.exe C:\User
s\usuario\Operadoreslogicos.c
C:\Users\usuario\Operadoreslogicos.c: In function 'main':
C:\Users\usuario\Operadoreslogicos.c:10:5: warning: assignment to 'char' from 'c
har *' makes integer from pointer without a cast [-Wint-conversion]
    c1 = "h";
    ^
C:\Users\usuario\Operadoreslogicos.c:11:5: warning: assignment to 'char' from 'c
har *' makes integer from pointer without a cast [-Wint-conversion]
    c2 = "H";
    ^
C:\Users\usuario\Operadoreslogicos.c:17:26: warning: comparison between pointer
and integer
    res = num1 < num2 && c1 == "h";
                        ^~
C:\Users\usuario\Operadoreslogicos.c:18:42: error: expected '}' before 'h'
    printf("\n num1 < num2 Y c1 es igual a "h" ? -> \t%d\n", res);
                                           ^
C:\Users\usuario\Operadoreslogicos.c:20:11: warning: comparison between pointer
and integer
    res = c1 == "S" || c2 == "H"
           ^~
C:\Users\usuario\Operadoreslogicos.c:20:24: warning: comparison between pointer
and integer
    res = c1 == "S" || c2 == "H"
           ^~
C:\Users\usuario\Operadoreslogicos.c:20:30: error: expected ';' before 'printf'
    res = c1 == "S" || c2 == "H"
                                ^
    printf("\n c1 es igual a 'S' O c2 a 'H'? -> \t%d\n", res);
    ^
C:\Users\usuario\MinGW\bin>
```

## Programa modificado

```
C:\Users\usuario\Documents\Operadoreslogicos.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Operadoreslogicos.c x
1  #include <stdio.h>
2
3  int main() {
4
5      int num1, num2, res;
6      char c1, c2;
7
8      num1 = 7;
9      num2 = 15;
10     c1 = 'h';
11     c2 = 'H';
12
13     printf("¿ num1 es menor a num2 ? -> \t%d\n", num1 < num2);
14     printf("¿ c1 es igual a c2 ? -> \t%d\n", c1 == c2);
15     printf("¿ c1 es diferente a c2 ? -> \t%d\n", c1 != c2);
16
17     res = num1 < num2 && c1 == 'h';
18     printf("¿ num1 < num2 Y c1 es igual a 'h' ? -> \t%d\n", res);
19
20     res = c1 == 's' || c2 == 'H';
21     printf("¿ c1 es igual a 's' O c2 a 'H'? -> \t%d\n", res);
22
23     return 0;
24 }
```

Finalmente si se compilo correctamente y se corrió en la terminal obteniendo resultados positivos y negativos en 0 y 1.

```
C:\Windows\system32\cmd.exe

C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Operadoreslogicos.exe C:\Users\usuario\Operadoreslogicos.c

C:\Users\usuario\MinGW\bin>C:\Users\usuario\Operadoreslogicos.exe
¿ 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
¿ c1 es igual a 's' O c2 a 'H'? ->      1

C:\Users\usuario\MinGW\bin>
```

## CONCLUSIONES

Como conclusión me gustó mucho esta práctica pues ya fue el usar un lenguaje de programación como tal y probar si estaba correcta. Me costó un poco de trabajo el compilar en la terminal de Windows pero era la única herramienta con la que contaba por lo tanto al final me gusto ya que fue como un reto porque tuve que acudir a conocimientos previos de clases anteriores. Algo que también me llamo mucho la atención es que para preguntas efectivamente la computadora da el resultado solo con dos valores 1 o 0. Creo es una práctica muy funcional para entender más de las bases de programación.

## FUENTES:

<https://drive.google.com/drive/folders/1CHK1FYtNYPwYzyiTklozMiRzIma6Jv6A>

<http://usagui06.blogspot.com/2014/05/compilar-y-ejecutar-un-programa-en-c-en.html>

<http://lsi.vc.ehu.es/asignaturas/FdIc/labs/a1/htm/oper.html>