	Carátula para entrega de prácticas	
Facultad de Ingeniería		Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	ALEJANDRO ESTEBAN PIMENTEL ALARCON
<i>Asignatura:</i>	FUNDAMENTOS DE PROGRAMACIÓN
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	7
<i>Integrante(s):</i>	ROMERO ROJAS ANA CRISTINA
<i>No. de Equipo de cómputo empleado:</i>	Niger 28
<i>No. de Lista o Brigada:</i>	1147
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	JUEVES 3 DE OCTUBRE
<i>Observaciones:</i>	En cada uno de los códigos hace falta evidencia (capturas) de que lo hayas compilado y ejecutado correctamente, sin eso sería solo copiar de la presentación

CALIFICACIÓN: 7

FUNDAMENTOS DE LENGUAJE C

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.

CONCEPTOS.

- Lenguaje C. Es un lenguaje de programación y que es de propósito general, esto significa que puede ser usado tanto para el desarrollo de sistemas operativos como para programas científicos, programas de aplicación o programas de educación y juegos.

DESARROLLO

Existen diferentes tipos de variables para números reales:

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

La diferencia entre ellas es la precisión con la que muestran sus valores.

```
miprimerprog.c
1  int main ()
2  {
3      printf("Mi primer programa\n dd");
4
5      return 0;
6  }
7  int main() {
8      //variables enteras
9      short 5;
10     signed int 7;
11     unsigned long 9;
12
13     //Caracter
14     char j;
15
16     //Variables reales
17     float x;
18     double b;
19
20     return 0;
21 }
```

También es posible leer y mostrar valores de diferentes tipos de variables con formatos específicos como se muestra a continuación:

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

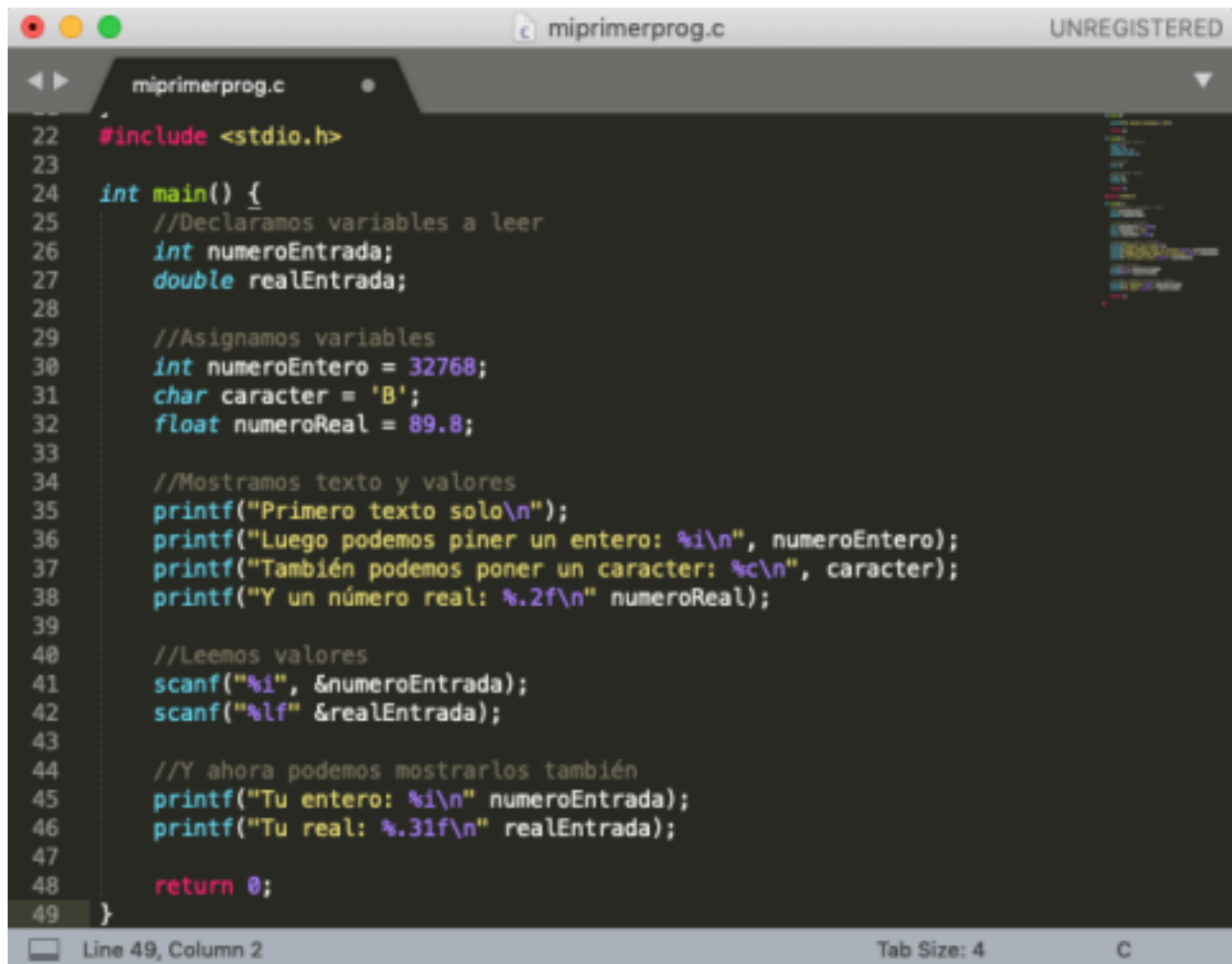
```

50  #include <stdio.h>
51
52  int main() {
53
54      int dos, tres, cuatro, cinco;
55      double fresultado;
56
57      dos = 2;
58      tres = 3;
59      cuatro = 4;
60      cinco = 5;
61
62      resultado = cinco/dos;
63      printf("5/2=%.11f\n", resultado);
64
65      resultado = (double)cinco/dos;
66      printf("5/2=%.11f\n", resultado);
67
68      return 0;
69  }
```

Line 69, Column 2

Tab Size: 4

C



```
22 #include <stdio.h>
23
24 int main() {
25     //Declaramos variables a leer
26     int numeroEntrada;
27     double realEntrada;
28
29     //Asignamos variables
30     int numeroEntero = 32768;
31     char caracter = 'B';
32     float numeroReal = 89.8;
33
34     //Mostramos texto y valores
35     printf("Primero texto solo\n");
36     printf("Luego podemos piner un entero: %i\n", numeroEntero);
37     printf("También podemos poner un caracter: %c\n", caracter);
38     printf("Y un número real: %.2f\n" numeroReal);
39
40     //Leemos valores
41     scanf("%i", &numeroEntrada);
42     scanf("%lf" &realEntrada);
43
44     //Y ahora podemos mostrarlos también
45     printf("Tu entero: %i\n" numeroEntrada);
46     printf("Tu real: %.3lf\n" realEntrada);
47
48     return 0;
49 }
```

Es posible también representar operaciones comunes que normalmente conocemos como suma, resta, multiplicación, etc. La manera correcta de mostrar los operadores es la siguiente:

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo

```

50 #include <stdio.h>
51
52 int main() {
53
54     int dos, tres, cuatro, cinco;
55     double fresultado;
56
57     dos = 2;
58     tres = 3;
59     cuatro = 4;
60     cinco = 5;
61
62     resultado = cinco/dos;
63     printf("5/2=%.11f\n", resultado);
64
65     resultado = (double)cinco/dos;
66     printf("5/2=%.11f\n", resultado);
67
68     return 0;
69 }

```

De igual manera podemos representar signos para comparar valores de dos variables así como signos de operadores lógicos:

<i>Uso</i>	<i>Resultado</i>
'h' == 'H'	Falso
'a' != 'b'	Verdadero
7 < 15	Verdadero
11 > 22	Falso
15 <= 22	Verdadero
20 >= 35	Falso

Como por ejemplo

<i>Operador</i>	<i>Operación</i>
==	Igual que
!=	Diferente a
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual

```
70 #include <stdio.h>
71
72 int main() {
73
74     int num1, num2, res;
75     char c1, c2;
76
77     num1 = 7;
78     num2 = 15;
79     c1 = 'h';
80     c2 = 'H';
81
82     printf("¿num1 es menor a num2? -> %d\n", num1<num2);
83     printf("¿c1 es igual a c2? -> %d\n", c1==c2);
84     printf("¿c1 es diferente a c2? -> %d\n", c1!=c2);
85
86     res = num1 < num2 && c1 == 'h';
87     printf("¿ num1 < num2 Y c1 es igual a 'h'? -> %d\n", res);
88
89     res = c1 == 's' || c2 == 'H';
90     printf("¿c1 es igual a 's' o c2 a 'H'? -> %d\n", res);
91
92     return 0;
93 }
```

Line 93, Column 2 Tab Size: 4 C

CONCLUSIÓN.

Es de suma importancia conocer cómo es posible declarar variables en lenguaje C así como representar de manera correcta diferentes símbolos para llevar a cabo diversas tareas y operaciones.