



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

### Laboratorios de computación salas A y B

<i>Profesor:</i>	M.C. 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>	Yuan Xiaojing
<i>No. de Equipo de cómputo empleado:</i>	35
<i>No. de Lista o Brigada:</i>	317693612
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	26-08-19
<i>Observaciones:</i>	Te faltan la compilación y ejecución del último programa, además, de que no se ejecutó correctamente el segundo

**CALIFICACIÓN:** 8

## Introducción

En ésta práctica tengo que Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos.

- Tipos de variables

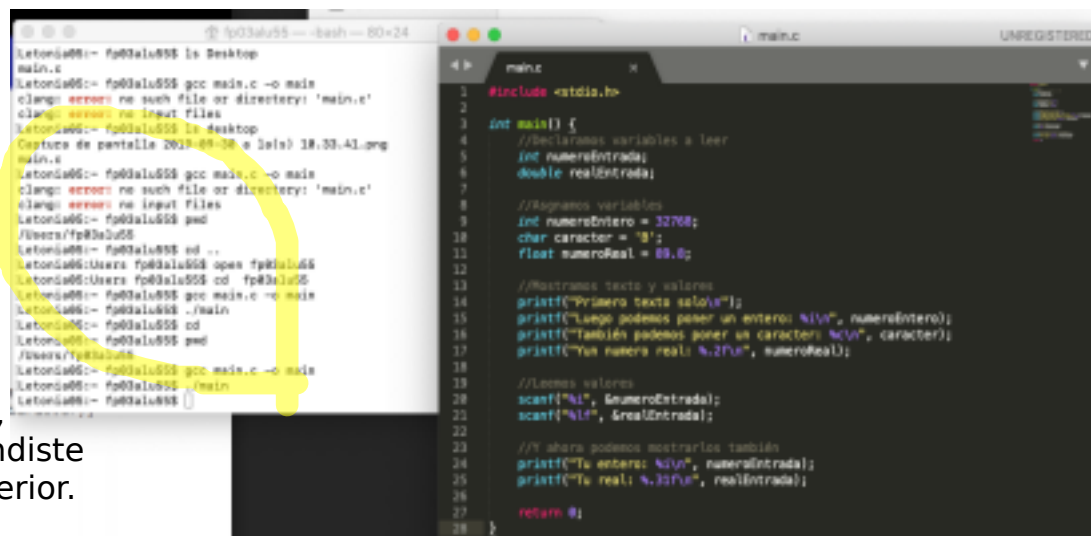
```
main.c
1 int main () {
2     // Variables enteras
3     short numeroEntero1;
4     signed int numeroEntero2;
5     unsigned long numeroEntero3;
6
7     //caracter
8     char caracter;
9
10    //Variables reales
11    float puntoFlotante1;
12    double puntoFlotante2;
13
14    return 0;
15 }
```

diferentes formas para expresar los tipos de variables.

compilar y correr el programa c para ver que sí hay errores o no

```
Letonia05:~ fp03alu55$ pwd
/Users/fp03alu55
Letonia05:~ fp03alu55$ cd ..
Letonia05:Users fp03alu55$ open fp03alu55
Letonia05:Users fp03alu55$ cd fp03alu55
Letonia05:~ fp03alu55$ gcc main.c -o main
Letonia05:~ fp03alu55$ ./main
Letonia05:~ fp03alu55$
```

- Mostrar y Leer



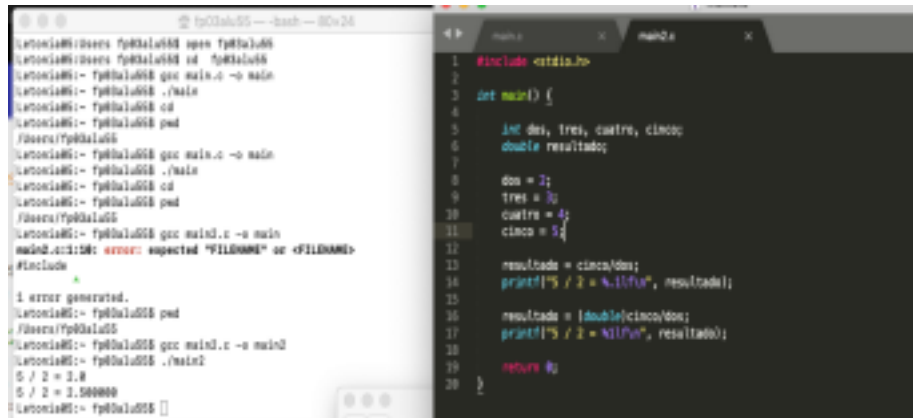
```
main.c
1 #include <stdio.h>
2
3 int main() {
4     //Declaramos variables a leer
5     int numeroEntero;
6     double realEntrada;
7
8     //Asignamos variables
9     int numeroEntero = 32768;
10    char caracter = 'a';
11    float numeroReal = 88.0;
12
13    //Mostramos texto y valores
14    printf("Primero texto solo\n");
15    printf("Luego podemos poner un entero: %i\n", numeroEntero);
16    printf("También podemos poner un caracter: %c\n", caracter);
17    printf("Y un numero real: %.2f\n", numeroReal);
18
19    //Leemos valores
20    scanf("%i", &numeroEntero);
21    scanf("%f", &realEntrada);
22
23    //Y ahora podemos mostrarlos también
24    printf("Te entero: %i\n", numeroEntero);
25    printf("Te real: %.2f\n", realEntrada);
26
27    return 0;
28 }
```

Primero hay que declarar los variables a leer que son números enteros y reales, luego asignar los variables que sean int, char o float, etc. Y ya puede mostrar txto y valores con el comando printf, para leer los valores usar scanf, en fin regresar con return 0.

aquí no está ocurriendo lo que debería ocurrir al ejecutar. Asegurate de que el archivo que estas compilando sea el correcto, quizá te confundiste y usaste el anterior.

- Operadores

Igual primero hay que declarar e asignar los valores, para mostrar los resultados usando los operadores para hacer las operaciones que quiera, y en fin regresar con return.



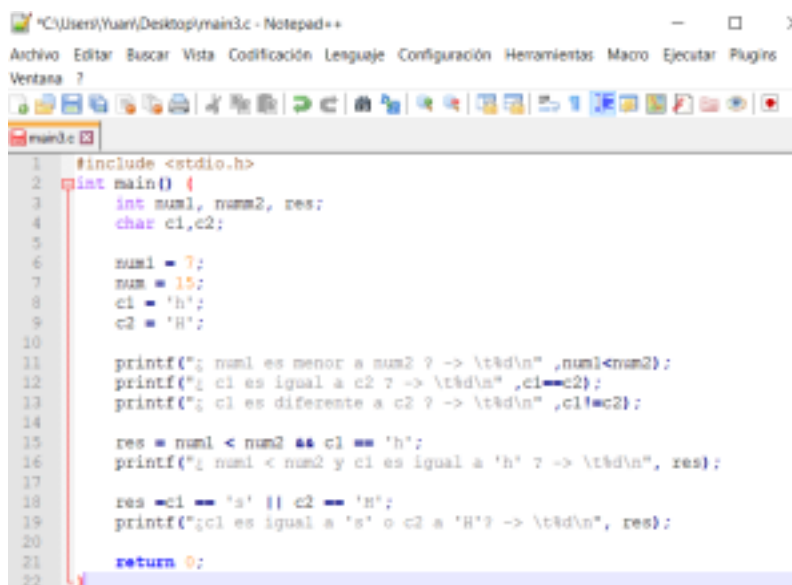
```

1 #include <stdio.h>
2
3 int main() {
4
5     int dos, tres, cuatro, cinco;
6     double resultado;
7
8     dos = 2;
9     tres = 3;
10    cuatro = 4;
11    cinco = 5;
12
13    resultado = cinco/dos;
14    printf("5 / 2 = %.1f\n", resultado);
15
16    resultado = (double)(cinco/tres);
17    printf("5 / 2 = %.1f\n", resultado);
18
19    return 0;
20 }

```

- Operadores lógicos

Después de declarar e asignar los valores, en donde mostrar los valores se puede compararlos y verificar si son verdaderos o falsos.



```

1 #include <stdio.h>
2 int main() {
3     int num1, num2, res;
4     char c1, c2;
5
6     num1 = 7;
7     num2 = 15;
8     c1 = 'h';
9     c2 = 'H';
10
11    printf("¿ num1 es menor a num2 ? -> %d\n", num1 < num2);
12    printf("¿ c1 es igual a c2 ? -> %d\n", c1 == c2);
13    printf("¿ c1 es diferente a c2 ? -> %d\n", c1 != c2);
14
15    res = num1 < num2 && c1 == 'h';
16    printf("¿ num1 < num2 y c1 es igual a 'h' ? -> %d\n", res);
17
18    res = c1 == 's' || c2 == 'H';
19    printf("¿ c1 es igual a 's' o c2 a 'H' ? -> %d\n", res);
20
21    return 0;
22 }

```

¿Dónde está la compilación y ejecución de este programa?

## Conclusión

El lenguaje C es un Lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas. Proporciona facilidades y libertades para realizar programas, y tiene compiladores para casi todos los sistemas conocidos.