	Carátula para entrega de prácticas	
Facultad de Ingeniería		Laboratorios de docencia

Laboratorio de Computación

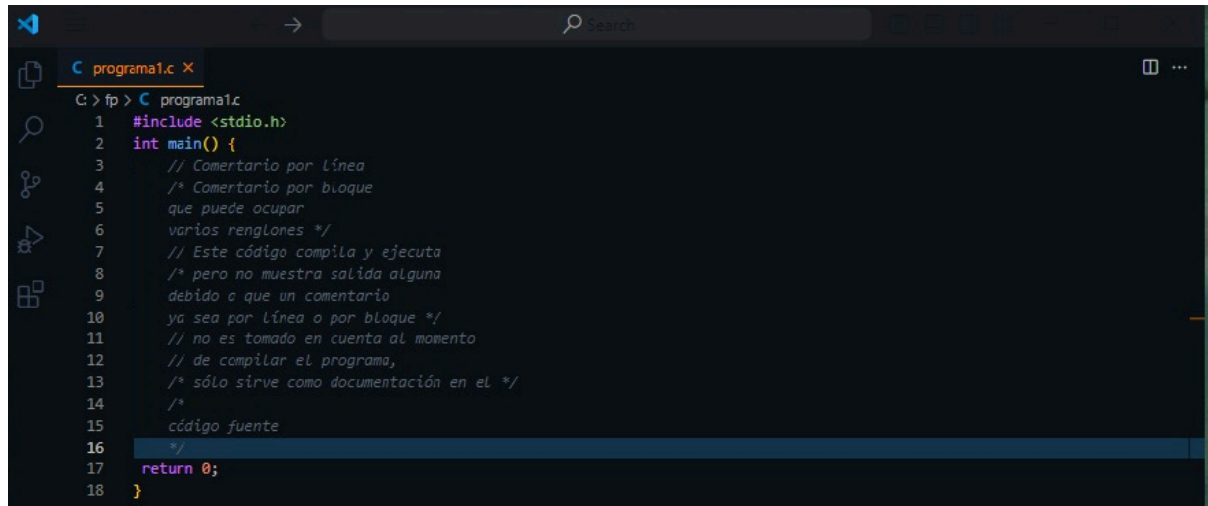
Salas A y B

Profesor(a):	César Fabián Domínguez Velasco
Asignatura:	Fundamentos de Programación
Grupo:	15
No de Práctica(s):	6
Integrante(s):	Camacho Duarte Héctor Enrique
	Gutiérrez Esquivel Giovani Emiliano
	Flores Jiménez Diego
	Moreno Chapan Amilet
	Reyes García Raúl de Jesús
No. de lista	06,11,15,16,25,33
o brigada:	
Semestre:	2024-02
Fecha de entrega:	20/03/24
Observaciones:	
CALIFICACIÓN:	

Objetivo:

El alumno elaborará 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.

Programa 1.c



```
C: > fp > C programa1.c
1  #include <stdio.h>
2  int main() {
3      // Comentario por línea
4      /* Comentario por bloque
5      que puede ocupar
6      varios renglones */
7      // Este código compila y ejecuta
8      /* pero no muestra salida alguna
9      debido a que un comentario
10     ya sea por línea o por bloque */
11     // no es tomado en cuenta al momento
12     // de compilar el programa,
13     /* sólo sirve como documentación en el */
14     /*
15     código fuente
16     */
17     return 0;
18 }
```

C:\ /cygdrive/c/FP

AMILET@DESKTOP-T56HH6F ~

\$ cd /cygdrive/c/FP

AMILET@DESKTOP-T56HH6F /cygdrive/c/FP

\$ ls

programa1.c

AMILET@DESKTOP-T56HH6F /cygdrive/c/FP

\$ gcc programa1.c -o programa1.exe

AMILET@DESKTOP-T56HH6F /cygdrive/c/FP

\$ _

Programa2.c

```
Welcome  C Programa2.c X
C Programa2.c
1  #include <stdio.h>
2
3  int main() {
4      short enteroNumero1 = 115;
5      signed int enteroNumero2 = 55;
6      unsigned long enteroNumero3 = 789;
7      char caracterA = 65;
8      char caracterB = 'B';
9      float puntoFlotanteNumero1 = 89.8;
10     //unsigned double puntoFlotanteNumero2 = 238.2236;
11     printf("%c\n", caracterA);
12     printf("%c\n", caracterB);
13     return 0;
14 }
```

```
gaeli@LAPTOP-VS6KK5M3 /home/fp
$ ls
Programa2.c

gaeli@LAPTOP-VS6KK5M3 /home/fp
$ gcc Programa2.c -o Programa2

gaeli@LAPTOP-VS6KK5M3 /home/fp
$ ls
Programa2.c  Programa2.exe

gaeli@LAPTOP-VS6KK5M3 /home/fp
$ ./Programa2.exe
A
B

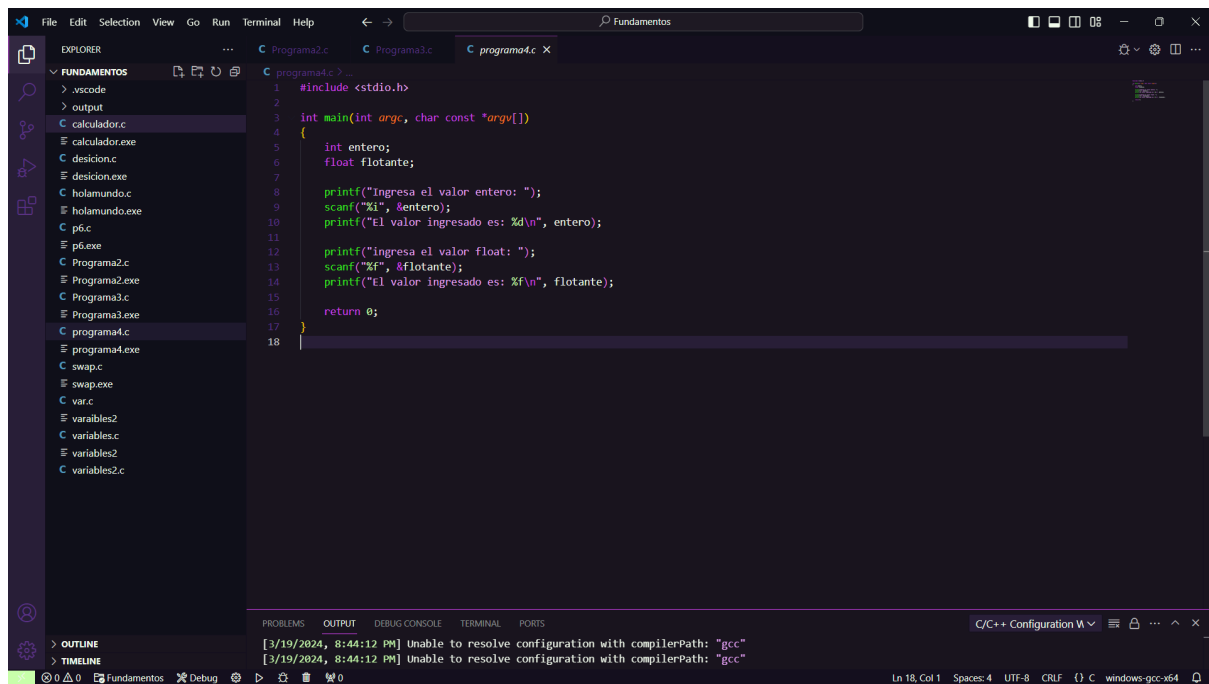
gaeli@LAPTOP-VS6KK5M3 /home/fp
$ |
```

Programa3.c

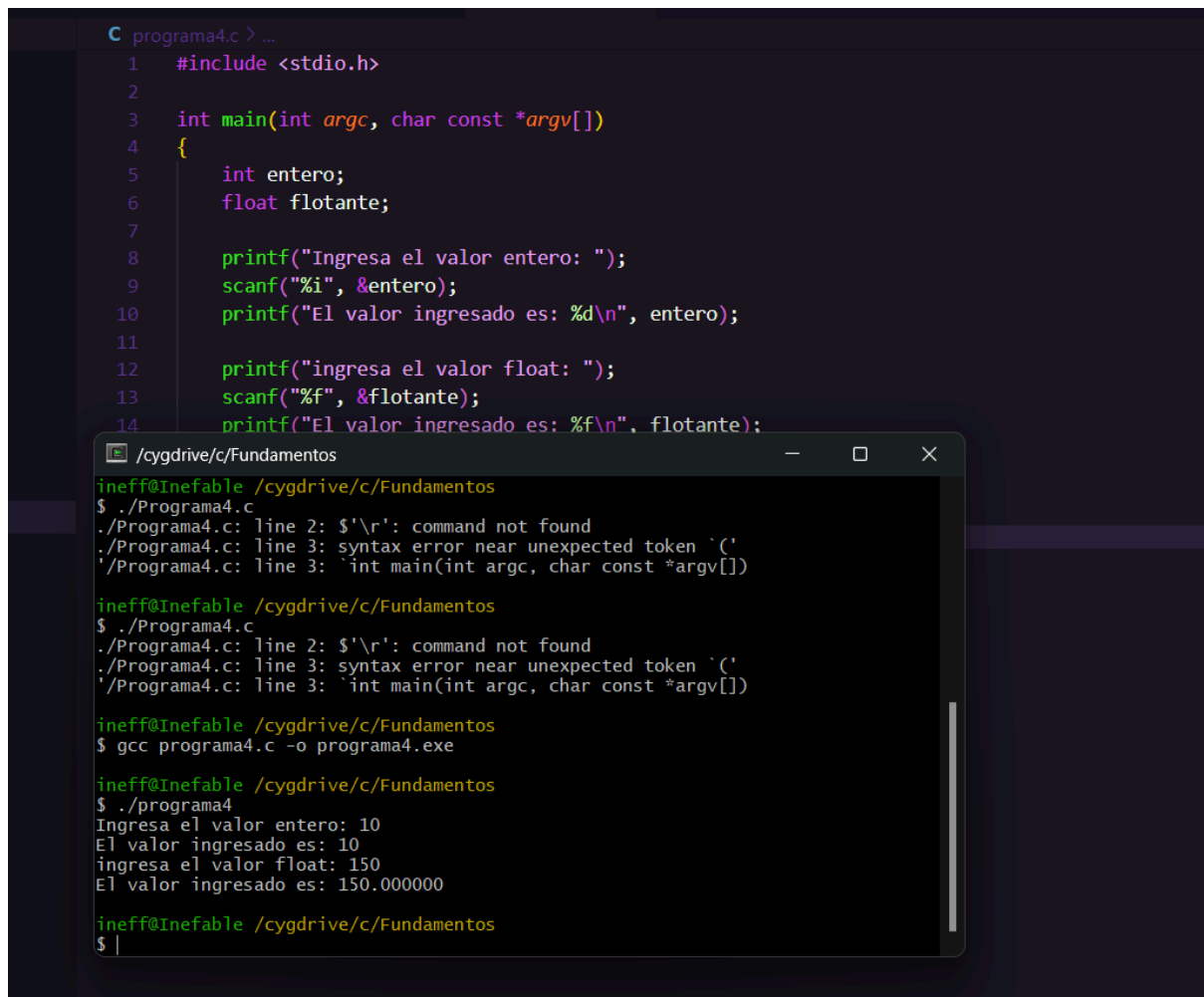
```
1  #include <stdio.h>
2
3  int main(int argc, char const *argv[])
4  {
5      // Declaración de variables
6      int entero;
7      float flotante;
8      double doble;
9      char caracter;
10     // Asignación de variables
11     entero = 14;
12     flotante = 3.5f;
13     doble = 6.8e10;
14     caracter = 'A';
15     // Funciones de salida de datos en pantalla
16
17     printf("La variable entera tiene valor : %i \n", entero);
18     printf("La variable flotante tiene valor : %f \n", flotante);
19     printf("La variable doble tiene valor : %d \n", doble);
20     printf("La variable caracter tiene valor : %c \n", caracter);
21     printf("Entero como octal : %o \n Como Hexadecimal %X \n", entero, entero);
22     printf("Flotante con precisión : %5.2f \n", flotante);
23     printf("Doble con precisión : %5.2f \n", doble);
24     printf("Carácter como entero : %d \n", caracter);
25     return 0;
26 }
```

```
/cygdrive/c/fp
Diego@DESKTOP-R3IB37S ~
$ cd /cygdrive/c/fp
Diego@DESKTOP-R3IB37S /cygdrive/c/fp
$ ls
programa3.c
Diego@DESKTOP-R3IB37S /cygdrive/c/fp
$ gcc programa3.c -o programa3
Diego@DESKTOP-R3IB37S /cygdrive/c/fp
$ ./programa3.exe
La variable entera tiene valor : 14
La variable flotante tiene valor : 3.500000
La variable doble tiene valor : 1342177280
La variable caracter tiene valor : A
Entero como octal : 16
Como Hexadecimal E
Flotante con precisión : 3.50
Doble con precisión : 680000000000.00
Carácter como entero : 65
Diego@DESKTOP-R3IB37S /cygdrive/c/fp
$ |
```

Programa4.c



```
1 #include <stdio.h>
2
3 int main(int argc, char const *argv[])
4 {
5     int entero;
6     float flotante;
7
8     printf("Ingresa el valor entero: ");
9     scanf("%i", &entero);
10    printf("El valor ingresado es: %d\n", entero);
11
12    printf("ingresa el valor float: ");
13    scanf("%f", &flotante);
14    printf("El valor ingresado es: %f\n", flotante);
15
16    return 0;
17 }
18
```



```
C programa4.c > ...
1 #include <stdio.h>
2
3 int main(int argc, char const *argv[])
4 {
5     int entero;
6     float flotante;
7
8     printf("Ingresa el valor entero: ");
9     scanf("%i", &entero);
10    printf("El valor ingresado es: %d\n", entero);
11
12    printf("ingresa el valor float: ");
13    scanf("%f", &flotante);
14    printf("El valor ingresado es: %f\n", flotante);
15
16    return 0;
17 }
18
```

```
/cygdrive/c/Fundamentos
ineff@Inefable /cygdrive/c/Fundamentos
$ ./Programa4.c
./Programa4.c: line 2: $'\r': command not found
./Programa4.c: line 3: syntax error near unexpected token `('
./Programa4.c: line 3: `int main(int argc, char const *argv[])

ineff@Inefable /cygdrive/c/Fundamentos
$ ./Programa4.c
./Programa4.c: line 2: $'\r': command not found
./Programa4.c: line 3: syntax error near unexpected token `('
./Programa4.c: line 3: `int main(int argc, char const *argv[])

ineff@Inefable /cygdrive/c/Fundamentos
$ gcc programa4.c -o programa4.exe

ineff@Inefable /cygdrive/c/Fundamentos
$ ./programa4
Ingresa el valor entero: 10
El valor ingresado es: 10
ingresa el valor float: 150
El valor ingresado es: 150.000000

ineff@Inefable /cygdrive/c/Fundamentos
$
```

Programa5.c

```
C ola.c  C Programa5.c X
C Programa5.c > ...
1  #include<stdio.h>
2  int main(int argc, char const *argv[]){
3  int enteroNumero;
4  char caracterA = 65; // Convierte el entero a carácter ASCII.
5  double puntoFlotanteNumero;
6  // Asignar valor de teclado a una variable.
7  printf("Escriba un valor entero:");
8  scanf("%i", &enteroNumero);
9  printf("Escriba un valor real:");
10 scanf("%lf", &puntoFlotanteNumero);
11 // Imprimir valores con formato.
12 printf("\nImprimiendo las variables enteras \a\n");
13 printf("\t Valor de enteroNumero = %i \a\n", enteroNumero);
14 printf("\t Valor de caracterA = %c \a\n", caracterA);
15 printf("\t Valor de puntoFlotanteNumero = %lf \a\n",
16 puntoFlotanteNumero);
17 printf("\t Valor de enteroNumero en base 16 = %x \a\n", enteroNumero);
18 printf("\t Valor de caracterA en código hexadecimal = %x\n", caracterA);
19 printf("\t Valor de puntoFlotanteNumero\n");
20 printf("en notación científica = %e\n", puntoFlotanteNumero);
21 return 0;
22 }
23
```

```
/cygdrive/c/FP
bpsom@Hope ~
$ cd /cygdrive/c/FP

bpsom@Hope /cygdrive/c/FP
$ ls
Programa5.c  ola.c  ola.exe

bpsom@Hope /cygdrive/c/FP
$ gcc Programa5.c -o Programa5.exe

bpsom@Hope /cygdrive/c/FP
$ ls
Programa5.c  Programa5.exe  ola.c  ola.exe

bpsom@Hope /cygdrive/c/FP
$ ./Programa5.exe
```

```
bpsom@Hope /cygdrive/c/FP
$ ./Programa5.exe
Escriba un valor entero:15
Escriba un valor real:3.14

Imprimiendo las variables enteras
    Valor de enteroNumero = 15
    Valor de caracterA = A
    Valor de puntoFlotanteNumero = 3.140000
    Valor de enteroNumero en base 16 = f
    Valor de caracterA en código hexadecimal = 41
    Valor de puntoFlotanteNumero
en notación científica = 3.140000e+00
```

Programa6.c

```
C Programa6.c > main()
1  #include<stdio.h>
2  int main(){
3
4      short ocho, cinco, cuatro, tres, dos, uno;
5
6      // 8 en binario: 0000 0000 0000 1000
7      ocho = 8;
8      // 5 en binario: 0000 0000 0000 0101
9      cinco = 5;
10     // 4 en binario: 0000 0000 0000 0100
11     cuatro = 4;
12     // 3 en binario: 0000 0000 0000 0011
13     tres = 3;
14     // 2 en binario: 0000 0000 0000 0010
15     dos = 2;
16     // 1 en binario: 0000 0000 0000 0001
17     uno = 1;
18     printf("Operadores aritméticos\n");
19     printf("5 modulo 2 = %d\n",cinco%dos);
20     printf("Operadores lógicos\n");
21     printf("8 >> 2 = %d\n",ocho>>dos);
22     printf("8 << 1 = %d\n",ocho<<1);
23     printf("5 & 4 = %d\n",cinco&cuatro);
24     printf("3 | 2 = %d\n",tres|dos);
25
26     printf("\n");
27     return 0;
28 }
```

```
/cygdrive/c/FP
bpsom@Hope /cygdrive/c/FP
$ cd ..
bpsom@Hope /cygdrive/c
$ cd ..
bpsom@Hope /cygdrive
$ cd ..
bpsom@Hope /
$ pwd
/
bpsom@Hope /
$ ls
Cygwin-Terminal.ico  Cygwin.ico  cygdrive  etc  lib  sbin  usr
Cygwin.bat          bin        dev      home  proc  tmp   var
bpsom@Hope /
$ cd /cygdrive/c/FP/
bpsom@Hope /cygdrive/c/FP
$ ls
Programa5.c  Programa5.exe  Programa6.c  ola.c  ola.exe
bpsom@Hope /cygdrive/c/FP
$ gcc Programa6.c -o Programa6.exe
bpsom@Hope /cygdrive/c/FP
$ ls
Programa5.c  Programa5.exe  Programa6.c  Programa6.exe  ola.c  ola.exe
bpsom@Hope /cygdrive/c/FP
$ ./Programa6.exe
Operadores aritméticos
5 modulo 2 = 1
Operadores lógicos
8 >> 2 = 2
8 << 1 = 16
5 & 4 = 4
3 | 2 = 3
```

Conclusión

Consideramos que hemos cumplido con los objetivos propuestos. Reforzamos lo aprendido en la práctica 2 al utilizar los comandos básicos del sistema operativo GNU/Linux para cambiar de directorios, visualizar nuestra ubicación actual y observar los archivos en la carpeta o directorio en el que nos encontramos. Principalmente, utilizamos estos conocimientos como herramienta para compilar códigos de los programas que desarrollamos en lenguaje C. Empleamos instrucciones de control secuencial para declarar variables y otros tipos de datos. Además, realizamos llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

Bibliografía

Laboratorio Salas A y B. (n.d.). <http://lcp02.fi-b.unam.mx/>

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.