

	<b>Carátula para entrega de prácticas</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>	3
<i>No de Práctica(s):</i>	7

<i>Integrante(s):</i>	Villanueva Bustamante Victoria
<i>No. de Equipo de cómputo empleado:</i>	19
<i>No. de Lista o Brigada:</i>	8034
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	30
<i>Observaciones:</i>	

CALIFICACIÓN:

---

## INTRODUCCIÓN

Uno de los objetivos de diseño del lenguaje C es que solo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución. Es muy posible escribir C a bajo nivel de abstracción; de hecho, C se usó como intermediario entre diferentes lenguajes.

## 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.

## ACTIVIDADES

```
programa.c  UNREGISTERED

1  int main()
2  {
3
4      // Variables enteras
5      short numeroEntero1;
6      signed int numeroEntero2;
7      unsigned long numeroEntero3;
8
9      // Caracter
10     char caracter;
11
12     // Variables reales
13     float puntoflotante1;
14     double puntoflotante2;
15
16     return 0;
17 }
```

Line 17, Column 2 Tab Size: 4 C

```
fp03alu54 ~ -bash — 80x24
Last login: Mon Sep 30 09:06:14 on tty000
Mayotte19:~ fp03alu54$ ls
Desktop      Downloads  Library    Music       Public
Documents    FP_2020-1_8034  Movies     Pictures    programa.c
Mayotte19:~ fp03alu54$ gcc programa.c -o nuevo
programa.c:13:2: warning: implicitly declaring library function 'printf' with
type 'int (const char *, ...)' [-Wimplicit-function-declaration]
printf("Primero texto solo\n");
^
programa.c:13:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
programa.c:16:31: warning: invalid conversion specifier '\x0a'
[-Wformat-invalid-specifier]
printf("Y un numero real: %.2\n", numeroReal);
^
programa.c:19:2: warning: implicitly declaring library function 'scanf' with
type 'int (const char *restrict, ...)' [-Wimplicit-function-declaration]
scanf("%i", &numeroEntrada);
^
programa.c:19:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'scanf'
3 warnings generated.
Mayotte19:~ fp03alu54$
```

```
programa.c  UNREGISTERED

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

Line 27, Column 2 Tab Size: 4 C

The top-left terminal window shows the execution of a C program. The user runs `gcc untitled.c -o prodos`. The output includes a warning about the implicit declaration of `printf` and a note to include `<stdio.h>`. The program calculates `5/2` and prints the result as `2.5`.

```
Mayotte19:~ fp03alu54$ gcc untitled.c -o prodos
untitled.c:12:2: warning: implicitly declaring library function 'printf' with
      type 'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("5/2 = %.1lf\n", resultado);
      ^
untitled.c:12:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Mayotte19:~ fp03alu54$
```

The top-right code editor shows the source code of `untitled.c`:

```
1 int main() {
2     int dos, tres, cuatro, cinco;
3     double resultado;
4
5     dos = 2;
6     tres = 3;
7     cuatro = 4;
8     cinco = 5;
9
10    resultado = cinco/dos;
11    printf("5/2 = %.1lf\n", resultado);
12
13    return 0;
14 }
```

The bottom-left terminal window shows the execution of a more complex C program. The user runs `gcc untitled.c -o cosa`. The output includes a warning about the implicit declaration of `printf` and a note to include `<stdio.h>`. The program prints several lines of text based on comparisons between variables `num1`, `num2`, `c1`, and `c2`.

```
Mayotte19:~ fp03alu54$ gcc untitled.c -o cosa
untitled.c:11:2: warning: implicitly declaring library function 'printf' with
      type 'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("%i num1 es menor a num2 ? -> %td\n", num1<num2);
      ^
untitled.c:11:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Mayotte19:~ fp03alu54$
```

The bottom-right code editor shows the source code of `untitled.c`:

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

## Conclusión

Esta práctica nos ayudó a conocer mejor el cómo se maneja el programador en C, que sinceramente se me hace más fácil que el pseudocódigo ya que se usa color.