	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de Programacion

Grupo: 3

No de Práctica(s): 9

Integrante(s): Velasco Gomez Noe Abimael

No. de Equipo de cómputo empleado: 32

No. de Lista o Brigada: 3989

Semestre: 2020-1

Fecha de entrega: 14/10/19

Observaciones:

CALIFICACIÓN: _____

Objetivo: Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

Ya hemos visto diferentes funciones por las cuales hemos podido crear funciones que cumplan con su tarea, sin embargo hay ciertos programas que no se pueden resolver por medio de las funciones vistas en otras prácticas, por lo tanto en esta practica se veran funciones de carácter ciclica, pudiendo así crear programas con resultados un poco más complejos y variados.

Actividades:
1)Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10).

Programa:

tdemultiplicar.c — UNREGISTERED

```c
int main ()
{
    //variables a leer//
    int x, y, n, resultado;

    //leer valores//
    scanf("%i", &x);

    //programa//
    n = 1;
    while (y!=resultado){
        y = x * n;
        n = n + 1;
        printf("Tabla: %i\n", y);
    resultado = x * 10;
    }

    return 0;
}
```

Ejecución:

```
Last login: Mon Oct 14 09:52:57 on ttys000
Monaco21:~ fp03alu53$ cd Downloads/
Monaco21:Downloads fp03alu53$ ls
ejemplo1.c              tlo                     tmo
tdemultiplicar.c        tm                      tmul
tkm                     tml
Monaco21:Downloads fp03alu53$ gcc tdemultiplicar.c -o tmlt
tdemultiplicar.c:7:2: warning: implicitly declaring library function 'scanf'
      with type 'int (const char *restrict, ...)'
      [-Wimplicit-function-declaration]
        scanf("%i", &x);
        ^
tdemultiplicar.c:7:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'scanf'
tdemultiplicar.c:14:3: warning: implicitly declaring library function 'printf'
      with type 'int (const char *, ...)' [-Wimplicit-function-declaration]
                printf("Tabla: %i\n", y);
                ^
tdemultiplicar.c:14:3: note: include the header <stdio.h> or explicitly provide
      a declaration for 'printf'
2 warnings generated.
Monaco21:Downloads fp03alu53$ ./tmlt
2
Tabla: 2
Tabla: 4
Tabla: 6
Tabla: 8
Tabla: 10
Tabla: 12
Tabla: 14
Tabla: 16
Tabla: 18
Tabla: 20
Monaco21:Downloads fp03alu53$
```

2)Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.

Programa:

sumaypromedio.c

```c
#include<stdio.h>

int main(){
    int i, suma = 0, n, prom;
    printf("digite la cantidad de numeros a sumar:\n");

    i = 1;

    while(i <= 10){
        scanf("%i", &n)

        suma = suma + n;
        i++;
    }
    prom = suma/10;
    printf("la suma es: %i", suma);
    printf("el promedio es: %i\n", prom);

    return 0;
}
```

Ejecución:

```
La suma es: 44
El promedio es: 4
julio@julio-VirtualBox ~/Descargas $ gcc su6.c -o su
julio@julio-VirtualBox ~/Descargas $ ./su
digite 10 numeron:
4
5
2
5
5
2
3
4
6
7
La suma es: 43
El promedio es: 4
julio@julio-VirtualBox ~/Descargas $ _
```

3)Hacer un programa que pida un número e indique si es primo o no.

Programa:

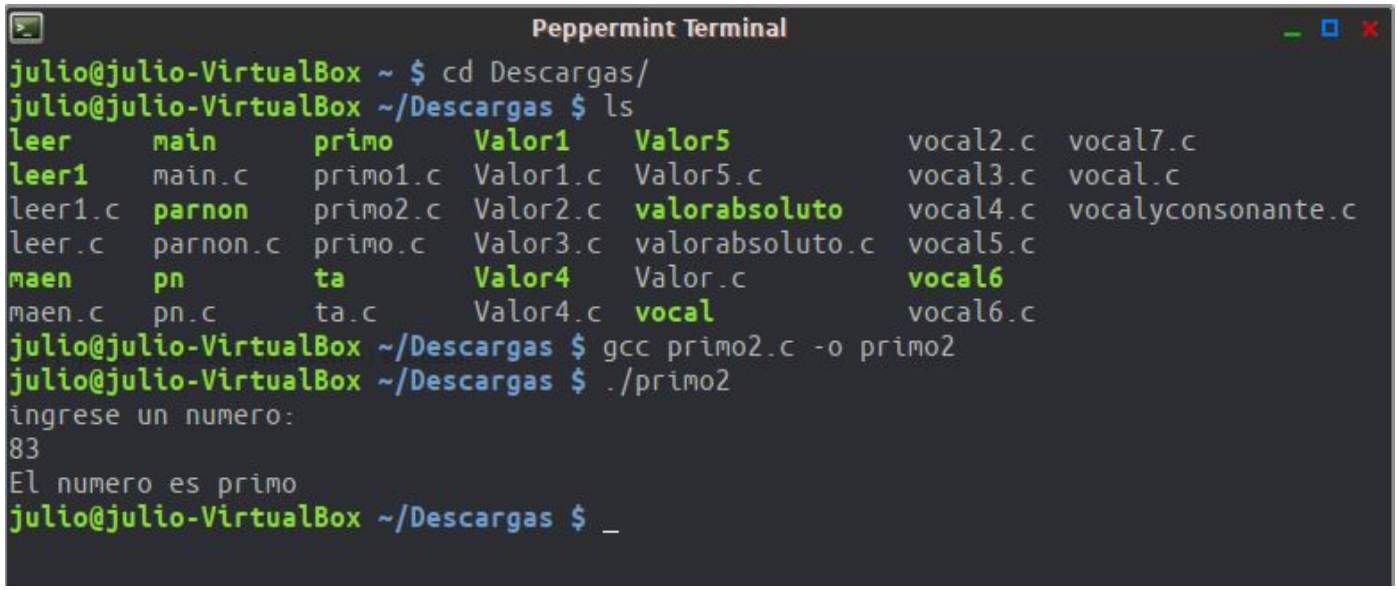
```c
#include<stdio.h>

int main(){
    int n, x, cont = 0;
    float operacion:
    printf("ingrese numero:\n");
    scanf("%i, &n");
    for(x = 1; x <= n; x++){
        if(n%x == 0){
            cont = cont + 1;
        }
    }
    if(cont > 2){
        printf("el numero es compuesto");
    }
    else{
        printf("el numero es primo");
    }

    return 0;
}
```

Line 21, Column 2 Tab Size: 4 C

Ejecución:



```
Peppermint Terminal
julio@julio-VirtualBox ~ $ cd Descargas/
julio@julio-VirtualBox ~/Descargas $ ls
leer      main      primo     Valor1    Valor5           vocal2.c  vocal7.c
leer1     main.c    primo1.c  Valor1.c  Valor5.c         vocal3.c  vocal.c
leer1.c   parnon    primo2.c  Valor2.c  valorabsoluto    vocal4.c  vocalyconsonante.c
leer.c    parnon.c  primo.c   Valor3.c  valorabsoluto.c  vocal5.c
maen      pn        ta        Valor4    Valor.c          vocal6
maen.c    pn.c      ta.c      Valor4.c  vocal            vocal6.c
julio@julio-VirtualBox ~/Descargas $ gcc primo2.c -o primo2
julio@julio-VirtualBox ~/Descargas $ ./primo2
ingrese un numero:
83
El numero es primo
julio@julio-VirtualBox ~/Descargas $ _
```

Para concluir las funciones DO, WHILE, FOR y demás vistas en esta práctica resultaron ser muy útiles para resolver tareas de mayor complejidad, sin embargo requieren de más especificaciones en los pasos que se quiera que se cuplan dentro del programa, concluyen que la utilización de estas funciones es un poco más compleja que el de las funciones de otras prácticas sobre lenguaje C, aunque la complejidad no es una mayor elevación de dificultad.