



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Karina García Morales

Profesor:

Fundamentos de programación

Asignatura:

20

Grupo:

Practica 8

No. de práctica(s):

Vargas Hernandez Edgar Vicente

Integrante(s):

50

No. de lista o

Primer semestre

Semestre:

22/11/2022

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo:

El alumno elaborará programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición.

Actividades:

- Elaborar un programa que utilice la estructura while en la solución de un problema
- Elaborar un programa que requiera el uso de la estructura do-while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while.
- Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while.

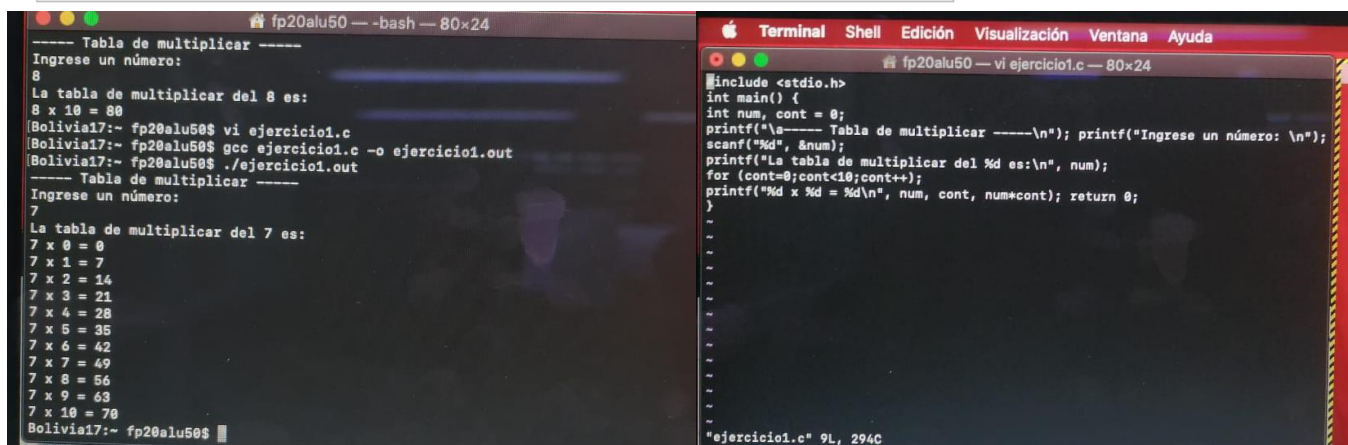
En la práctica número 8 la profesora nos dijo que íbamos a comenzar a ver el funcionamiento de la estructura while. La cual empieza validando la expresión lógica y si esta se cumple o es verdadera es cuando empieza a ejecutar el bloque de instrucciones de la estructura al igual que los demás estructuras está delimitada por las llaves que cierran los programas pero cuando esta condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura dando paso directo a la siguiente fase o estructura.

```
while (expresión_lógica) {  
    // Bloque de código a repetir  
    // mientras que la expresión  
    // lógica sea verdadera.  
}
```

Como en todas las prácticas de laboratorio comenzamos haciendo el programa 1 donde la profesora nos explica que con la estructura while se puede generar una tabla de multiplicar la cual nosotros le otorgamos un número y el programa ejecutará la tabla de multiplicar.

Programa1.c

```
#include <stdio.h>  
  
int main()  
{  
    int num, cont = 0;  
    printf("\a----- Tabla de multiplicar ----- \n");  
    printf("Ingrese un número: \n");  
    scanf("%d", &num);  
    printf("La tabla de multiplicar del %d es:\n", num);  
    while (++cont <= 10)  
        printf("%d x %d = %d\n", num, cont, num*cont);  
    return 0;  
}
```



The image shows two side-by-side terminal windows. The left window displays the output of the program for the number 8, showing a multiplication table from 8x0 to 8x10. The right window shows the source code of the program being edited in a terminal window.

```
#fp20alu50 -bash - 80x24  
----- Tabla de multiplicar -----  
Ingrese un número:  
8  
La tabla de multiplicar del 8 es:  
8 x 0 = 80  
[Bolivia17:~ fp20alu50$ vi ejercicio1.c  
[Bolivia17:~ fp20alu50$ gcc ejercicio1.c -o ejercicio1.out  
[Bolivia17:~ fp20alu50$ ./ejercicio1.out  
----- Tabla de multiplicar -----  
Ingrese un número:  
7  
La tabla de multiplicar del 7 es:  
7 x 0 = 0  
7 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
7 x 4 = 28  
7 x 5 = 35  
7 x 6 = 42  
7 x 7 = 49  
7 x 8 = 56  
7 x 9 = 63  
7 x 10 = 70  
[Bolivia17:~ fp20alu50$
```

```
Terminal Shell Edición Visualización Ventana Ayuda  
#fp20alu50 - vi ejercicio1.c - 80x24  
#include <stdio.h>  
int main() {  
    int num, cont = 0;  
    printf("\a----- Tabla de multiplicar ----- \n"); printf("Ingrese un número: \n");  
    scanf("%d", &num);  
    printf("La tabla de multiplicar del %d es:\n", num);  
    for (cont=0;cont<10;cont++);  
    printf("%d x %d = %d\n", num, cont, num*cont); return 0;  
}
```

"ejercicio1.c" 9L, 294C

En el programa número dos al momento de yo realizar la práctica no noté algo significativo porque al momento de ejecutar el mismo programa no me hacía nada una vez que la profesión empezó a explicar que como era verdadero no sé realizaba la acción pues éste fue cuando entendí que en esencia no iba a ejecutar nada porque era verdadero el programa luego nos pidió cambiar el valor y cuando yo lo cambie Me empezó a escribir un montón de veces el mismo programa a lo cual se hizo infinito toda la parte que estaba encerrada entre paréntesis o entre comillas a lo que tuve que cerrar el programa y volverlo a abrir para poder seguir trabajando

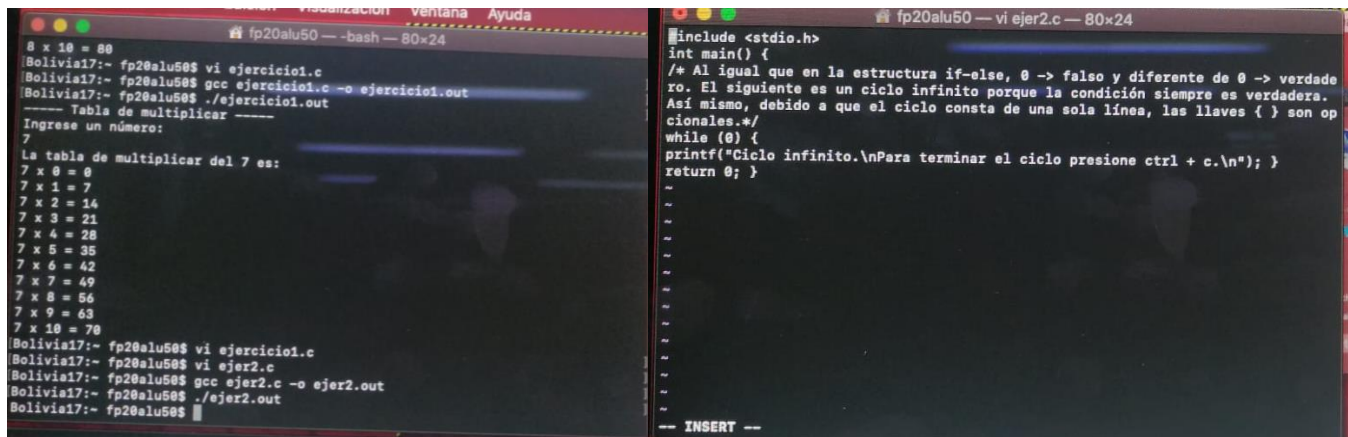
Programa2.c

```
#include <stdio.h>

int main()
{

    /* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero.
    El siguiente es un ciclo infinito porque la condición siempre es verdadera.
    Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son
    opcionales.*/

    while (100)
    {
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
    }
    return 0;
}
```



```
fp20alu50 ~ -bash- 80x24
8 x 10 = 80
Bolivia17:~ fp20alu50$ vi ejercicio1.c
Bolivia17:~ fp20alu50$ gcc ejercicio1.c -o ejercicio1.out
Bolivia17:~ fp20alu50$ ./ejercicio1.out
----- Tabla de multiplicar -----
Ingrese un número:
7
La tabla de multiplicar del 7 es:
7 x 0 = 0
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
Bolivia17:~ fp20alu50$ vi ejercicio1.c
Bolivia17:~ fp20alu50$ vi ejer2.c
Bolivia17:~ fp20alu50$ gcc ejer2.c -o ejer2.out
Bolivia17:~ fp20alu50$ ./ejer2.out
Bolivia17:~ fp20alu50$

fp20alu50 ~ vi ejer2.c - 80x24
#include <stdio.h>
int main() {
    /* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero. El siguiente es un ciclo infinito porque la condición siempre es verdadera. Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son opcionales.*/
    while (0) {
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
    }
    return 0;
}
-- INSERT --
```

Luego vimos la estructura do while En este primero ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición por lo que el bloque de código se ejecuta las veces que sean necesarias para poder pasar o validar y escribir lo que se tenga que escribir

```
do {
    /*
    Bloque de código que se ejecuta
    por lo menos una vez y se repite
    mientras la expresión lógica sea
    verdadera.
    */
} while (expresión_lógica);
```

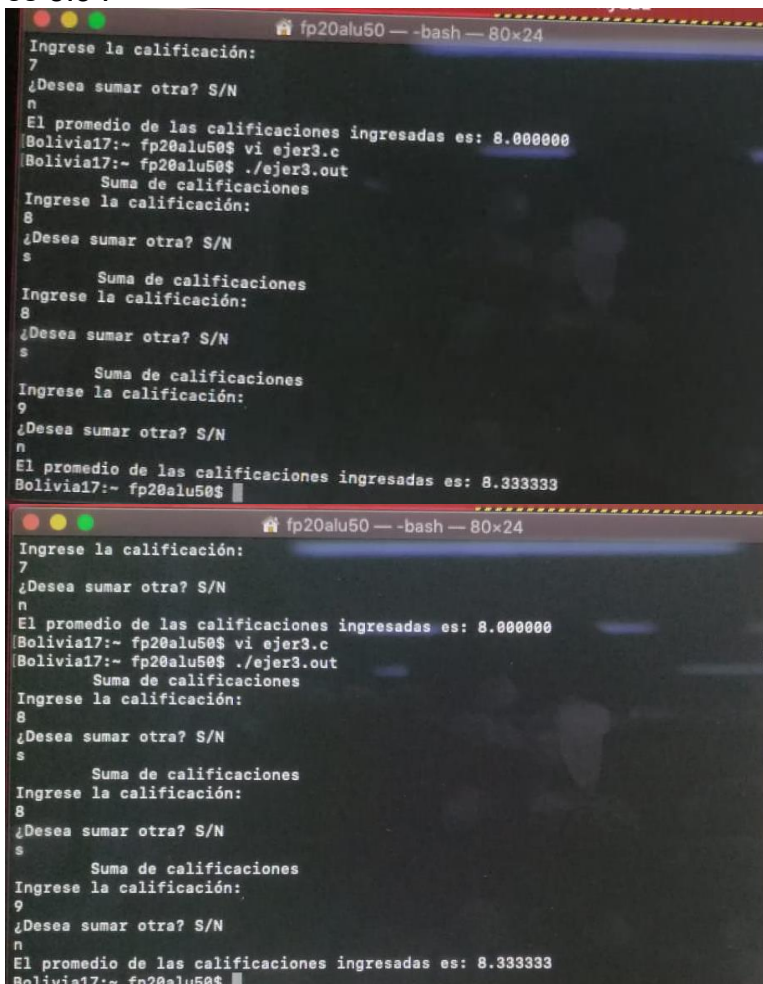
Un dato que nos dio la profesora y que está puesto en la práctica es que si nosotros colocamos solamente un bloque de código oh una sola condición nosotros podemos omitir el abrir y cerrar las llaves pero siempre tenemos que terminar con el punto y coma (;)

El siguiente programa el programa número 3 el objetivo es obtener el promedio de calificaciones ingresados por el usuario nosotros colocamos las calificaciones y de ahí seleccionamos una de las letras asignadas para poder pasar a escribir la siguiente calificación o cerrar el programa.

Programa3.c

```
#include <stdio.h>
int main ()
{
    char op = 'n';
    double sum = 0, calif = 0;
    int veces = 0;
    do
    {
        printf("\tSuma de calificaciones\n");
        printf("Ingrese la calificación:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;
        printf("¿Desea sumar otra? S/N\n");
        setbuf(stdin, NULL); // limpia el buffer del teclado
        scanf("%c",&op);
        getchar();
    }
    while (op == 'S' || op == 's');
    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
    return 0;
}
```

En mi caso yo coloque solamente cuatro calificaciones y de ahí cerre el programa que fueron 8 9 y 8 por lo que el programa me sacó como resultado el promedio de las calificaciones ingresadas es 8.3 .



```
fp20alu50 — -bash — 80x24
Ingrese la calificación:
7
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 8.000000
Bolivia17:~ fp20alu50$ vi ejer3.c
Bolivia17:~ fp20alu50$ ./ejer3.out
Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
9
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 8.333333
Bolivia17:~ fp20alu50$

fp20alu50 — -bash — 80x24
Ingrese la calificación:
7
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 8.000000
Bolivia17:~ fp20alu50$ vi ejer3.c
Bolivia17:~ fp20alu50$ ./ejer3.out
Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
9
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 8.333333
Bolivia17:~ fp20alu50$
```

En el siguiente programa el número cuatro la profesora nos dijo que era el programa para realizar una calculadora sencilla donde ocuparíamos la sintaxis de do lo único que nos pidió la profesora era colocar el programa ejecutarlo y realizar una de las cuatro opciones que estaban disponibles como suma resta multiplicación y división.

Programa4.c

```
#include <stdio.h>
int main ()
{
    int op, uno, dos;
    do
    {
        printf(" --- Calculadora ---\n");
        printf("\n¿Qué desea hacer\n");
        printf("1) Sumar\n");
        printf("2) Restar\n");
        printf("3) Multiplicar\n");
        printf("4) Dividir\n");
        printf("5) Salir\n");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                printf("\tSumar\n");
                printf("Introduzca los números a sumar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d + %d = %d\n", uno, dos, (uno + dos));
                break;
            case 2:
                printf("\tRestar\n");
                printf("Introduzca los números a restar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d - %d = %d\n", uno, dos, (uno - dos));
                break;
            case 3:
                printf("\tMultiplicar\n");
                printf("Introduzca los números a multiplicar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d * %d = %d\n", uno, dos, (uno * dos));
                break;
            case 4:
                printf("\tDividir\n");
                printf("Introduzca los números a dividir separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
                break;
            case 5:
                printf("\tSalir\n");
                break;
            default:
                printf("\tOpción inválida.\n");
        }
    }
    while (op != 5);
}
```

```
Bolivia17:~ fp20alu50$ vi ejer4.c
Bolivia17:~ fp20alu50$ gcc ejer4.c -o ejer4.out
Bolivia17:~ fp20alu50$ ./ejer4.out
--- Calculadora ---
¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
2
Restar
Introduzca los números a restar separados por comas
257,86
257 - 86 = 171
--- Calculadora ---
¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
Salir
Bolivia17:~ fp20alu50$
```

```
#include <stdio.h>
int main () {
    int op, uno, dos; do
    {
        printf(" --- Calculadora ---\n");
        printf("\n¿Qué desea hacer\n");
        printf("1) Sumar\n");
        printf("2) Restar\n");
        printf("3) Multiplicar\n");
        printf("4) Dividir\n");
        printf("5) Salir\n");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                printf("\tSumar\n");
                printf("Introduzca los números a sumar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d + %d = %d\n", uno, dos, (uno + dos));
                break;
            case 2:
                printf("\tRestar\n");
                printf("Introduzca los números a restar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d - %d = %d\n", uno, dos, (uno - dos));
                break;
            case 3:
                printf("\tMultiplicar\n");
                printf("Introduzca los números a multiplicar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d * %d = %d\n", uno, dos, (uno * dos));
                break;
            case 4:
                printf("\tDividir\n");
                printf("Introduzca los números a dividir separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
                break;
            case 5:
                printf("\tSalir\n");
                break;
            default:
                printf("\tOpción inválida.\n");
        }
    }
    while (op != 5);
}
```

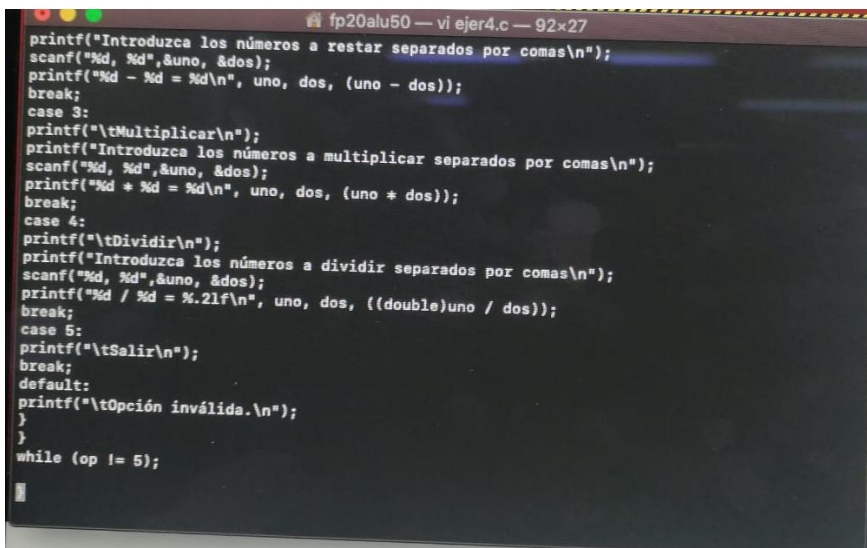

Luego vimos otra estructura la cual era de repetición nombrada como “for” la cual como su nombre lo dice permite realizar repeticiones cuando generalmente el control de la repetición está definida sobre una variable contador esta es su sintaxis

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {  
    /*  
        Bloque de código  
        a ejecutar  
    */  
}
```

En el programa número 5 encontramos otra forma para poder tener las calificaciones pero en vez de un solo alumno de varios alumnos por lo que al igual que en el anterior programa se realiza un promedio de dichas calificaciones.

Programa5.c

```
#include <stdio.h>  
int main ()  
{  
    int enteroNumAlumnos = 5;  
    float realCalif = 0.0, realPromedio = 0.0;  
    printf("\tPromedio de calificaciones\n");  
    for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)  
    {  
        printf("\nIngrese la calificación del alumn %d\n", indice+1);  
        scanf("%f",&realCalif);  
        realPromedio += realCalif;  
    }  
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",  
        realPromedio/enteroNumAlumnos);  
    return 0;  
}
```



```
fp20alu50 — vi ejer4.c — 92x27  
printf("Introduzca los números a restar separados por comas\n");  
scanf("%d, %d",&uno, &dos);  
printf("%d - %d = %d\n", uno, dos, (uno - dos));  
break;  
case 3:  
printf("\tMultiplicar\n");  
printf("Introduzca los números a multiplicar separados por comas\n");  
scanf("%d, %d",&uno, &dos);  
printf("%d * %d = %d\n", uno, dos, (uno * dos));  
break;  
case 4:  
printf("\tDividir\n");  
printf("Introduzca los números a dividir separados por comas\n");  
scanf("%d, %d",&uno, &dos);  
printf("%d / %d = %.21f\n", uno, dos, ((double)uno / dos));  
break;  
case 5:  
printf("\tSalir\n");  
break;  
default:  
printf("\tOpción inválida.\n");  
}  
while (op != 5);
```

```
fp20alu50 -- -bash -- 92x27
4) Dividir
5) Salir
5
Salir
Bolivia17:~ fp20alu50$ vi ejer4.c
Bolivia17:~ fp20alu50$ vi ejer5.c
Bolivia17:~ fp20alu50$ gcc ejer5.c -o ejer5.out
Bolivia17:~ fp20alu50$ ./ejer5.out
Promedio de calificaciones
Ingrese la calificación del alumn 1
9
Ingrese la calificación del alumn 2
7
Ingrese la calificación del alumn 3
10
Ingrese la calificación del alumn 4
7
Ingrese la calificación del alumn 5
8
El promedio de las calificaciones ingresadas es: 8.200000
Bolivia17:~ fp20alu50$
```

En el último programa se define un valor por defecto para el tamaño del arreglo este programa es la verdad no le entendí mucho pero utilizaba la estructura Max algo que la verdad si me quedó bastante claro con lo que nos mencionó la profesora Max obtiene un valor determinado dependiendo de donde esté colocado por lo que nosotros al colocarle el Max con el número 5 íbamos a tener 5 apartados donde íbamos a poder colocarle una variable

Programa5.c

```
#include <stdio.h>
#define MAX 5
int main ()
{
    int arreglo[MAX], cont;
    for (cont=0; cont<MAX; cont++)
    {
        printf("Ingrese el valor %d del arreglo: ", cont+1);
        scanf("%i", &arreglo[cont]);
    }

    printf("El valor ingresado para cada elemento del arreglo es:\n[");
    for (cont=0; cont<MAX; cont++)
    {
        printf("%d\t", arreglo[cont]);
    }
    printf("]\n");
    return 0;
}
```

```
fp20alu50 -- -bash -- 92x27
Ingrese la calificación del alumn 1
9
Ingrese la calificación del alumn 2
7
Ingrese la calificación del alumn 3
10
Ingrese la calificación del alumn 4
7
Ingrese la calificación del alumn 5
8
El promedio de las calificaciones ingresadas es: 8.200000
Bolivia17:~ fp20alu50$ vi ejer6.c
Bolivia17:~ fp20alu50$ gcc ejer6.c -o ejer6.out
Bolivia17:~ fp20alu50$ ./ejer6.out
Ingrese el valor 1 del arreglo: 1
Ingrese el valor 2 del arreglo: 8
Ingrese el valor 3 del arreglo: 4
Ingrese el valor 4 del arreglo: 9
Ingrese el valor 5 del arreglo: 7
El valor ingresado para cada elemento del arreglo es:
[1      8      4      9      7      ]
Bolivia17:~ fp20alu50$
```

```
fp20alu50 -- vi ejer6.c -- 92x27
#include <stdio.h>
#define MAX 4
#define p printf
#define s scanf
int main ()
{
    int arreglo[MAX], cont;
    for (cont=0; cont<MAX; cont++)
    {
        p("Ingrese el valor %d del arreglo: ", cont+1);
        s("%i", &arreglo[cont]);
    }
    p("El valor ingresado para cada elemento del arreglo es:\n[");
    for (cont=0; cont<MAX; cont++)
    {
        p("%d\t", arreglo[cont]);
    }
    p("]\n");
    return 0;
}
-- INSERT --
```

```
fp20alu50 — vi ejer6.c — 92x27
#include <stdio.h>
#define MAX 4
#define p printf
#define s scanf
int main ()
{
int arreglo[MAX], cont;
for (cont=0; cont<MAX; cont++)
{
p("Ingrese el valor %d del arreglo: ", cont+1);
s("%i", &arreglo[cont]);
}
p("El valor ingresado para cada elemento del arreglo es:\n");
for (cont=0; cont<MAX; cont++)
{
p("%d\t", arreglo[cont]);
}
p("\n");
return 0;
}
~
~
~
~
-- INSERT --
```

Tarea:

1.- Solicitar radio a usuario y calcular el área.

```
#include <stdio.h>
int main()
{
float area, radio;
printf("\n introduce el radio del circulo");
scanf("%f", &radio);
area=3.1416*radio;
printf("\n El area es. %5F", area);
return 0;
}
~
"ejer1.c" [New] 10L, 184B written
[root@localhost ~]# gcc ejer1.c -o ejer1.out
[root@localhost ~]# ./ejer1.out
\n introduce el radio del circulo10
\n El area es. 31.416000[root@localhost ~]#
```

2.- Mostrar la numeración del 10 al 1 haciendo uso del ciclo while

```
#include <stdio.h>
int main()
{
float area, radio;
printf("\n introduce el radio del circulo");
#include <stdio.h>
int main()
{
int contador=10;
while( contador>0)
{
printf("%d", contador),contador = contador-1;
#include <stdio.h>
int main()
{
int contador=10;
while( contador>0)
{
printf("%d", contador),contador = contador-1;
}
return 0;
}
}
~
"ejer2.c" 10L, 130B written
[root@localhost ~]# gcc ejer2.c -o ejer2.out
[root@localhost ~]# ./ejer2.out
10987654321[root@localhost ~]#
```


3.- Mostrar la tabla de multiplicar del número que ingrese el usuario con el ciclo for.

```
#include <stdio.h>
int main()
{
    int num, cont = 0;
    printf("a-----Tabla de multiplicar-----\n");
    printf("Ingresar un numero: |n");
    #include <stdio.h>
    int main()
    {
        int num, cont = 0;
        printf("-----Tabla de multiplicar-----");
        printf("Ingresar un numero: |n");
        scanf("%d", &num);
        printf("La tabla de multiplicar del %d es:", num);
        for(cont=0; ++cont <=10;)
            printf("%d*%d=%d", num, cont, num*cont);
        return 0;
    }
}

"ejer3.c" 12L, 281B written
[root@localhost ~]# gcc ejer3.c -o ejer3.out
[root@localhost ~]# ./ejer3.out
-----Tabla de multiplicar-----Ingresar un numero: |n8
La tabla de multiplicar del 8 es:8*1=88*2=168*3=248*4=328*5=408*6=488*7=568*8=64
8*9=728*10=80[root@localhost ~]#
```

Conclusion:

La práctica en sí estuvo bastante sencilla porque no tuvimos que realizar demasiados cambios en las estructuras pero sí se notó cierta complejidad al momento de realizar los mismos programas IA que ahora con un solo valor que este escrito puede cambiar cualquier resultado que se nos dé en el caso donde nosotros colocamos un número diferente tanto el 5 como el número cuatro en el último ejercicio alado del Max que muestra un diferente resultado en cada 1 de ellos por lo que creo que la práctica fue sencilla pero demasiado compleja.

https://github.com/Chente117/practica1_fbp.git