



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina Morales Garcia

Asignatura: Fundamentos de programacion

Grupo: 20

No. de práctica(s): 7

Integrante(s): Avila Pineda Samuel David

No. de lista o brigada: 06

Semestre: 2023-1

Fecha de entrega: 20 de noviembre del 2022

Observaciones:

CALIFICACIÓN: _____

Objetivo

El alumno elaborará programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Actividades

- Elaborar expresiones lógicas/condicionales utilizadas en las estructuras de selección y realizar su evaluación.
- Elaborar un programa en lenguaje C para cada estructura de selección.

Introducción

¿Qué hacen las estructuras de control de flujo?

Es un lenguaje especifican el orden en que se realiza un procesamiento de datos.

¿Qué hacen las estructuras de selección?

También son conocidas como condicionales y nos permiten realizar una u otra acción con base en una expresión lógica. Las acciones solo se pueden ejecutar una a la vez dentro de toda la estructura.

¿Cuántas estructuras de selección posee el lenguaje C?

Son 3, las cuales son:

- If-else
- Switch
- Condicional o ternaria.

A continuación, se describirá cada una de ellas y se mostraran algunos ejemplos, con las evidencias del día de la práctica.

Estructura de control selectiva if.

Esta es la estructura más simple, su sintaxis es:

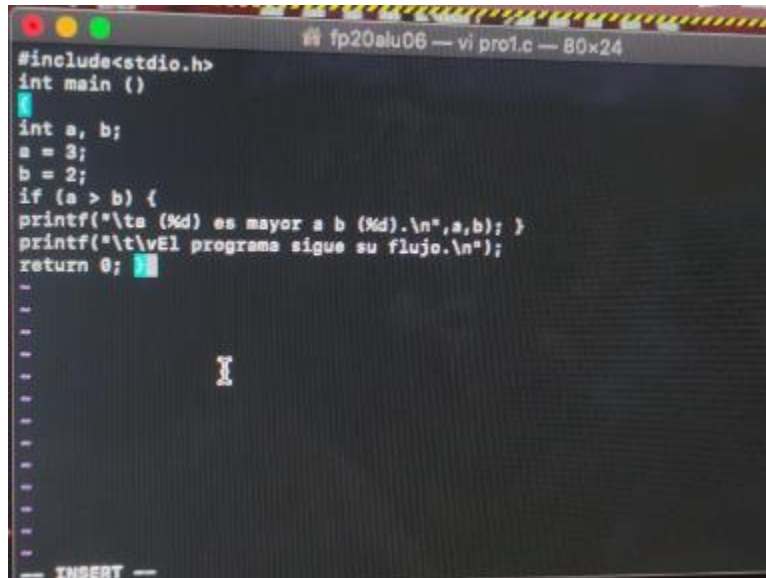
```
if (expresión_lógica)
{
    // bloque de código a ejecutar
}
```

Primero se evalúa la expresión lógica y si es verdadera, posteriormente se ejecutan las instrucciones, que se encuentran entre las llaves, de lo contrario sino se cumple se continua con el flujo normal del programa.

Programa 1

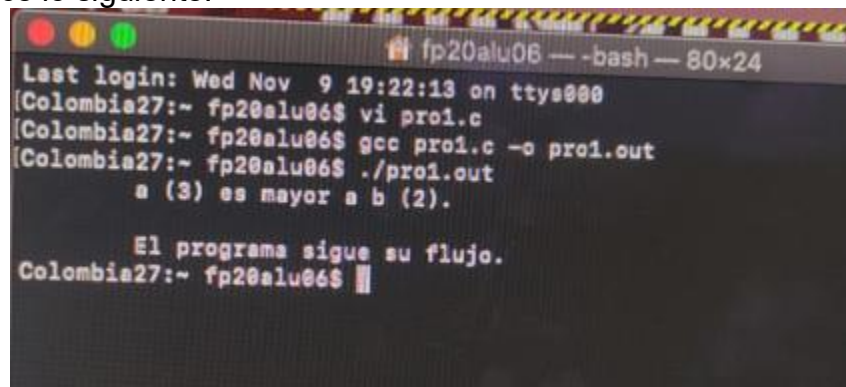
Este es el primer programa que realizamos.

```
Programa1.c
#include<stdio.h>
int main ()
{
    int a, b;
    a = 3;
    b = 2;
    if (a > b)
    {
        printf("\ta (%d) es mayor a b (%d).\n",a,b);
    }
    printf("\t\vEl programa sigue su flujo.\n");
    return 0;
}
```



```
#include<stdio.h>
int main ()
{
    int a, b;
    a = 3;
    b = 2;
    if (a > b) {
        printf("\ta (%d) es mayor a b (%d).\n",a,b); }
    printf("\t\tvEl programa sigue su flujo.\n");
    return 0; }
```

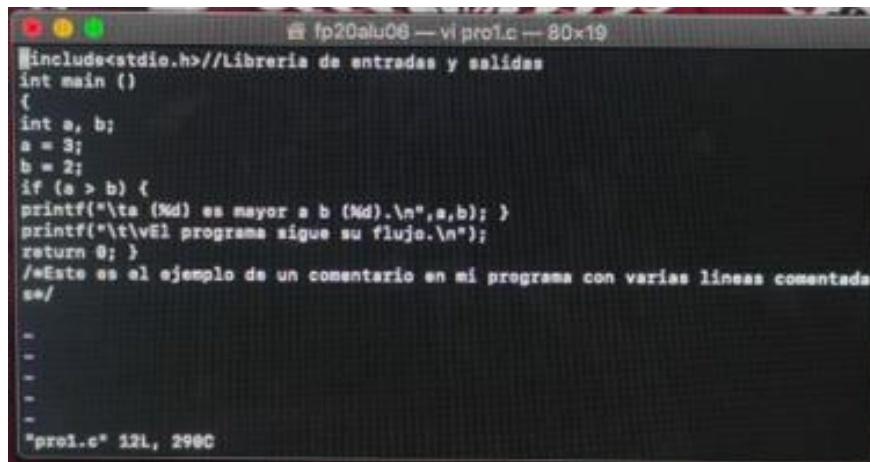
Como se puede ver, abrimos una terminal en nuestra computadora, posteriormente creamos un archivo, yo le coloqué el nombre de “pro1.c”. Me salí de ese archivo y después hice lo siguiente:



```
Last login: Wed Nov  9 19:22:13 on ttys000
[Colombia27:~ fp20alu06$ vi pro1.c
[Colombia27:~ fp20alu06$ gcc pro1.c -o pro1.out
[Colombia27:~ fp20alu06$ ./pro1.out
    a (3) es mayor a b (2).

    El programa sigue su flujo.
Colombia27:~ fp20alu06$
```

Se compiló el programa, como vimos que no se encontraba ningún error en el programa, lo ejecute y todo salió bien. Después de esto, la profesora nos dijo que en este programa teníamos que colocar algunos comentarios. Como sabemos, para colocar comentarios usamos la siguiente sintaxis “// el comentario que uno quiera poner”.



```
#include<stdio.h> //Librería de entradas y salidas
int main ()
{
    int a, b;
    a = 3;
    b = 2;
    if (a > b) {
        printf("\ta (%d) es mayor a b (%d).\n",a,b); }
    printf("\t\tvEl programa sigue su flujo.\n");
    return 0; }
/*Este es el ejemplo de un comentario en mi programa con varias líneas comentadas*/

"pro1.c" 12L, 290C
```

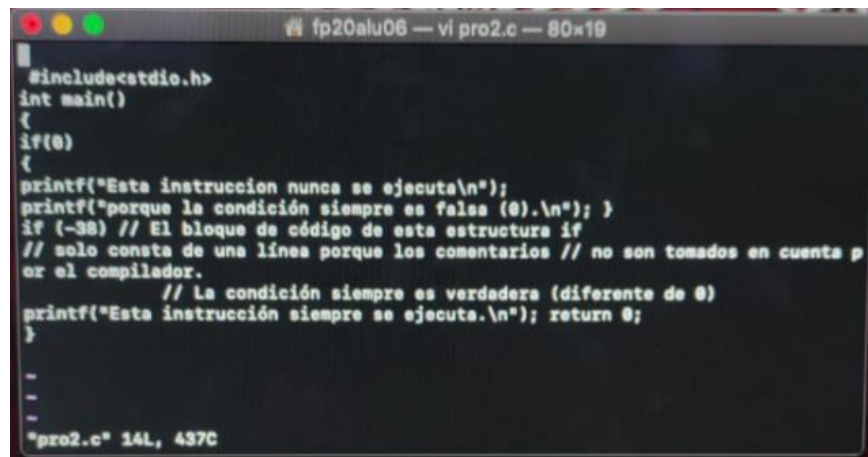
Yo coloque en el primer renglón el comentario.

Después de esto se volvió a compilar y ejecutar.

Como podemos ver se comprueba que las condiciones son numéricas 0 –falso y diferente de 0 verdadero.

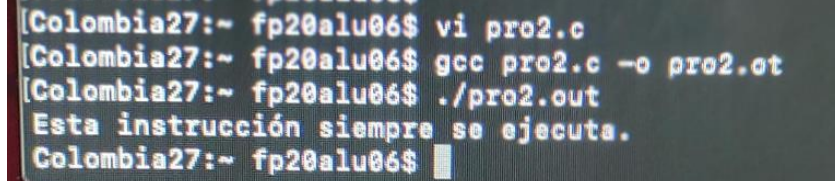
Programa 2

```
#include<stdio.h>
int main()
{
    if (0)
    {
        printf("Esta instrucción nunca se ejecuta\n");
        printf("porque la condición siempre es falsa (0).\n");
    }
    if (-38) // El bloque de código de esta estructura if
            // solo consta de una línea porque los comentarios
            // no son tomados en cuenta por el compilador.
            // La condición siempre es verdadera (diferente de 0)
        printf("Esta instrucción siempre se ejecuta.\n");
    return 0;
}
```



```
#include<stdio.h>
int main()
{
    if(0)
    {
        printf("Esta instrucción nunca se ejecuta\n");
        printf("porque la condición siempre es falsa (0).\n"); }
    if (-38) // El bloque de código de esta estructura if
            // solo consta de una línea porque los comentarios // no son tomados en cuenta p
            or el compilador.
            // La condición siempre es verdadera (diferente de 0)
        printf("Esta instrucción siempre se ejecuta.\n"); return 0;
}
~
~
~
"pro2.c" 14L, 437C
```

Como lo hicimos anteriormente, primero se copió el programa, en este se puede apreciar que muestran diferentes comentarios, no lo modificamos ni nada.



```
Colombia27:~ fp20alu06$ vi pro2.c
Colombia27:~ fp20alu06$ gcc pro2.c -o pro2.ot
Colombia27:~ fp20alu06$ ./pro2.out
Esta instrucción siempre se ejecuta.
Colombia27:~ fp20alu06$
```

Aquí se compilo y podemos ver que no hubo ningún problema, por eso mismo se ejecutó.

```
fp20alu06 — vi pro2.c — 80x19
#include<stdio.h>
int main()
{
    if(0)
    {
        printf("Esta instruccion nunca se ejecuta\n");
        printf("porque la condición siempre es falsa (0).\n"); }
    if (3) // El bloque de código de esta estructura if
        // solo consta de una línea porque los comentarios // no son tomados en cuenta p
        or el compilador.
        // La condición siempre es verdadera (diferente de 0)
        printf("Esta instrucción siempre se ejecuta.\n"); return 0;
    }
}

~
~
~
"pro2.c" 14L, 435C
```

La profesora nos dijo que cambiáramos el valor de "-38" por otro, yo decidí cambiarlo por "3", esto con el fin de ver qué pasaba.

```
Colombia27:~ fp20alu06$ vi pro2.c
Colombia27:~ fp20alu06$ gcc pro2.c -o pro2.out
Colombia27:~ fp20alu06$ ./pro2.out
Esta instrucción siempre se ejecuta.
Colombia27:~ fp20alu06$
```

Al igual que como la primera vez nos imprimió "Esta instrucción siempre se ejecuta"

Programa Edad

La profesora nos dejó un ejercicio que teníamos que hacer en ese momento, era solicitar la edad al usuario e indicar si era mayor de edad o no.

```
fp20alu06 — vi edad1.c — 80x19
#include<stdio.h>
int main()
{
    int EDAD;
    printf("¿Cual es tu edad?\n");
    scanf("%d",&EDAD);
    if (EDAD>=18){
        printf("Eres mayor de edad\n");
    }
    else {
        printf("No eres mayor de edad\n");
        return 0;
    }
}

~
~
~
"edad1.c" 16L, 199C
```

Primero abrí un archivo con el nombre de "edad1.c" después ya hice el código fuente del ejercicio que nos pidió.

```
Colombia27:~ fp20alu06$ vi edad1.c
Colombia27:~ fp20alu06$ gcc edad1.c -o edad1.out
Colombia27:~ fp20alu06$ ./edad1.out
¿Cual es tu edad?
19
Eres mayor de edad
Colombia27:~ fp20alu06$
```

Se compilo y no mostro ningún error, se ejecutó, coloque mi edad "19" y me imprimió que soy mayor de edad.

Estructura de control selectiva if-else

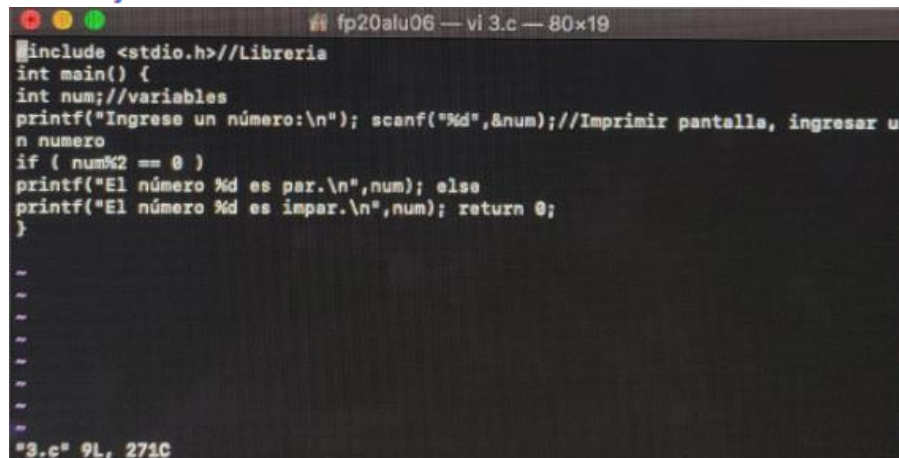
La sintaxis es:

```
if (expresión_lógica)
{
    // bloque de código a ejecutar
    // si la condición es verdadera
}
else
{
    // bloque de código a ejecutar
    // si la condición es falsa
}
```

En esta se evalúa la expresión lógica, si es verdadera, se ejecutan las instrucciones del bloque que se encuentra en las primeras llaves, sino se ejecuta el bloque que está entre las llaves después de la palabra else.

Programa 3

```
#include <stdio.h>
int main()
{
    int num;
    printf("Ingrese un número:\n");
    scanf("%d",&num);
    if ( num%2 == 0 )
        printf("El número %d es par.\n",num);
    else
        printf("El número %d es impar.\n",num);
    return 0;
}
```



Cree un archivo con el nombre "pro3.c", después copie y pegue en la terminal el código fuente que nos dieron en la práctica, La profesora nos dijo que fuéramos colocando comentarios, yo coloque en el primer renglón y en el cuarto. Además, se cambiaron las comillas, ya que si lo compilábamos así nos iba a marcar error.


```

[Colombia27:~ fp20alu06$ vi 3.c
[Colombia27:~ fp20alu06$ gcc 3.c -o 3.out
[Colombia27:~ fp20alu06$ ./3.out
Ingrese un número:
12
El número 12 es par.
Colombia27:~ fp20alu06$

```

Se compilo el programa y no marco errores, después lo ejecute y nos pidió un número, a la hora de darle enter nos iba a imprimir si es un numero par o impar, en este caso yo coloque el 12 y me imprimió que es par.

Programa 4

```

#include <stdio.h>
int main()
{
    int uno, dos, tres;
    printf ("Ingrese 3 números separados por espacios:\n");
    scanf ("%d %d %d", &uno, &dos, &tres);
    if (uno > dos)
    {
        if (dos > tres)
        {
            printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres);
        }
        else
        {
            if (uno > tres)
            {
                printf("%d es mayor a %d que es mayor a %d\n", uno, tres, dos);
            }
            else
            {
                printf("%d es mayor a %d que es mayor a %d\n", tres, uno, dos);
            }
        }
    }
    else
    {
        if (dos > tres)
        {
            if (tres > uno)
            {
                printf("%d es mayor a %d que es mayor a %d\n", dos, tres, uno);
            }
            else
            {
                printf("%d es mayor a %d que es mayor a %d\n", dos, uno, tres);
            }
        }
        else
        {
            printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
        }
    }
    return 0;
}

```

```

#include <stdio.h>
int main() {
    int uno, dos, tres;
    printf ("Ingrese 3 números separados por espacios:\n"); scanf ("%d %d %d", &uno, &dos, &tres);
    if (uno > dos)
    {
        if (dos > tres) {
            printf ("%d es mayor a %d que es mayor a %d\n", uno, dos, tres); }
        else
        {
            if (uno > tres)
            {
                printf ("%d es mayor a %d que es mayor a %d\n", uno, tres, dos);
            }
            else
            {
                printf ("%d es mayor a %d que es mayor a %d\n", tres, uno, dos);
            }
        }
    }
    else
    {
        if (dos > tres)
        {
            if (tres > uno)
            {
                printf ("%d es mayor a %d que es mayor a %d\n", dos, tres, uno);
            }
            else
            {
                printf ("%d es mayor a %d que es mayor a %d\n", dos, uno, tres);
            }
        }
        else {
            printf ("%d es mayor a %d que es mayor a %d\n", tres, dos, uno); }
    }
    return 0; }

```

Cree un archivo con el nombre de “pro4.c”, en este no hice ninguna modificación solo copie y pegue.

```

Colombia27:~ fp20alu06$ vi pro4.c
Colombia27:~ fp20alu06$ gcc pro4.c -o pro4.out
Colombia27:~ fp20alu06$ ./pro4.out
Ingrese 3 números separados por espacios:
3 4 6
6 es mayor a 4 que es mayor a 3
Colombia27:~ fp20alu06$ █

```

Compile el programa y no mostro ningún error, ejecute y pues me pidió 3 números separados por espacios le di a la tecla enter y me puso que numero era mayor entre uno y otro.

Estructura de control selectiva switch-case

La sintaxis es la que se muestra a continuación:

```

switch (opcion_a_evaluar)
{
    case valor1:
        /* Código a ejecutar*/
        break;
    case valor2:
        /* Código a ejecutar*/
        break;
    case valorN:
        /* Código a ejecutar*/
        break;
    default:
        /* Código a ejecutar*/
}

```

En este evalúa la variable que se encuentra entre paréntesis después la palabra reservada switch y la comprara con los valores constantes que posee cada paso. Este puede evaluar datos como enteros, caracteres y enumeraciones. Al final de cada caso se ejecuta la instrucción break.

Si la opción a evaluar no coincide, se ejecuta el bloque default. Este por lo general se escribe al final. Es importante que si se coloca en otra parte del bloque debe

¿Para qué se utiliza el break?

Programa 5

```
#include <stdio.h>
int main(){
char op = '\0';
printf("\tMenú\n\n");
printf("Elegir la opción deseada\n"); printf("a) jugar voleibol\n");
printf("b) jugar basquetbol\n"); printf("c) practicar natacion\n"); scanf("%c",&op);
switch(op)
{
default:
printf("Opción no válida.\n"); break;
case 'a':
printf("Se seleccionó 'jugar voleibol'.\n"); break;
case 'b':
printf("Se seleccionó 'jugar basquetbol'.\n"); break;
case 'c':
printf("Se seleccionó 'practicar natacion'.\n"); break;
}
return 0; }
```

```
--
--
--
--
--
--
```

```
twa
```

Igual que anteriormente, cree un archivo, este con el nombre de "pro5.c", la actividad de este era cambiar el menú, yo coloque jugar voleibol, jugar basquetbol o practicar natación.

```
Elegir la opción deseada
a) jugar voleibol
b) jugar basquetbol
c) practicar natacion
a
Se seleccionó 'jugar voleibol'.
Colombia27:~ fp20alu06$
```

Compile y ejecute el programa, todo salió bien, pues se puede ver que a la hora de ejecutar me arrojó un menú, que es el que yo coloque, seleccione una de las 3 opciones y me imprimió el resultado de lo que coloque

Enumeración

Una variable enumerador se puede crear de la siguiente manera:

```
enum identificador {VALOR1, VALOR2, ... , VALORN};
```

Como podemos observar para crear una enumeración se utiliza la palabra enum, seguida de un identificador (nombre), entre llaves se ingresan los nombres de los valores que puede tomar dicha enumeración. Se escriben en mayúsculas debido a que son elementos enteros constantes.

La enumeración se llama boolean y contiene dos elementos (FALSE) posee el valor de 0 y (TRUE) posee el valor de 1.

Programa 8

```
#include <stdio.h>
int main()
{
    // Los valores de una enumeración son enteros y constantes
    enum díasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO};
    int op;
    printf("Ingrese el día de la semana.\n");
    printf("1) Lunes\n");
    printf("2) Martes\n");
    printf("3) Miércoles\n");
    printf("4) Jueves\n");

    printf("5) Viernes\n");
    printf("6) Sábado\n");
    printf("7) Domingo\n");
    scanf("%d", &op);
    switch(op-1)
    {
        case LUNES:
        case MARTES:
            printf("Inicio de semana.\n");
            break;
        case MIERCOLES:
            printf("Mitad de semana.\n");
            break;
        case JUEVES:
            printf("|Casi inicia el fin de semana!\n");
            break;
        case VIERNES:
        case SABADO:
            printf("|Fin de semana!\n");
            break;
        case DOMINGO:
            printf("Día de descanso.\n");
            break;
        // No se necesita default
    }

    return 0;        // Valor entero en hexadecimal
}
```

```

fp20alu06 — vi pro8.c — 121x48
#include <stdio.h>
int main() {
    // Los valores de una enumeración son enteros y constantes
    enum díasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO}; //se ennumeran los días de la semana
    int op;
    printf("Ingrese el día de la semana.\n");
    printf("1) Lunes\n");
    printf("2) Martes\n");
    printf("3) Miércoles\n");
    printf("4) Jueves\n");
    printf("5) Viernes\n");
    printf("6) Sábado\n");
    printf("7) Domingo\n");
    scanf("%d", &op);
    switch(op-1) //no entendi muy bien lo que significa el op-1
    {
        case LUNES:
        case MARTES:
            printf("Inicio de semana.\n");
            break;
        case MIERCOLES:
            printf("Mitad de semana.\n");
            break;
        case JUEVES:
            printf("¡Casi inicia el fin de semana!\n");
            break;
        case VIERNES:
        case SABADO:
            printf("¡Fin de semana!\n");
            break;
        case DOMINGO:
            printf("Día de descanso.\n");
            break;
        // No se necesita default
    }
    return 0; // Valor entero en hexadecimal
}

```

Se creó un archivo con el nombre de pro8.c, lo único que hice fue copiar y pegar, agregué un comentario en el renglón 15, no entendí lo que significaba el op-1.

```

[Colombia27:~ fp20alu06$ vi pro8.c
[Colombia27:~ fp20alu06$ gcc pro8.c -o pro8.out
[Colombia27:~ fp20alu06$ ./pro8.out
Ingrese el día de la semana.
1) Lunes
2) Martes
3) Miércoles
4) Jueves
5) Viernes
6) Sábado
7) Domingo
6
¡Fin de semana!

```

Compile y no salió ningún error, entonces ejecute y me apareció un menú de los días de la semana, coloque el número 6 y apreté la tecla enter, lo que me imprimió fue que es fin de semana.

Estructura de control selectiva condicional

Esta nos permite realizar una comparación rápida. La sintaxis es la siguiente:

Condición ? SiSeCumple : SiNoSeCumple

Consta de 3 partes, la condición y dos acciones a seguir. Si la condición se cumple se ejecuta la instrucción que se encuentra después del símbolo "?" sino se cumple se ejecuta la instrucción que se encuentra después de ":".

Programa 9

```

#include <stdio.h>
int main()
{
    double a, b, res;
    printf("Calcular el error matemático E = |a - b|\n\n");
    printf("Ingrese el valor de a:\n");
    scanf("%lf",&a);
    printf("Ingrese el valor de b:\n");
    scanf("%lf",&b);
    res = a < b ? b-a : a-b;
    printf("El error matemático de\n");
    printf("| %lf - %lf | es %lf\n", a, b, res);
    return 0;
}

```

```

[Colombia27:~ fp20alu06$ vi pro9.c
[Colombia27:~ fp20alu06$ gcc pro9.c -o pro9.out
[Colombia27:~ fp20alu06$ ./pro9.out
Calcular el error matemático E = |a - b|

Ingrese el valor de a:
9
Ingrese el valor de b:
7
El error matemático de
| 9.000000 - 7.000000 | es 2.000000
Colombia27:~ fp20alu06$

```

La única foto que encontré del último programa, es la que se muestra en la parte de arriba, las actividades que teníamos que hacer, es que teníamos que colocar comentarios, de los pasos que se muestran, después de esto era compilar y ejecutar.

A la hora de ejecutar, primero me pidió un valor, lo coloque y luego me pidió uno más, después de esto me arrojó el error matemático entre uno y otro.

Tarea

1.- Completar la tabla con diagrama, [pseudocódigo](#) y codificación para la estructura condicional, adjunto un ejemplo.

Estructura	Diagrama	Pseudocódigo	Codificación
Secuencial	<pre> graph TD A[/variableSum a: REAL/] --> B[variableSuma ← 0] B --> C[/variableSuma/] C --> D["El valor de la suma es: "variableSuma"] </pre>	Declarar variable Inicializar variable LEER ESCRIBIR	Float a; a=0; scanf ("%f", &a) printf ("el valor de la suma")

Estructura	Diagrama	Pseudocódigo	Codificación
Condicional simple	<pre> graph TD Cond{Condición} -- F --> Join1(()) Cond -- V --> Acciones[Acciones] Acciones --> Join2(()) Join1 --> Join2 </pre>	SI (condición) ENTONCES //Acciones FIN DEL SI	if (condición) { Acciones a realizar }
Condicional compuesto	<pre> graph TD Cond{Condición} -- F --> Acciones1[Acciones] Cond -- V --> Acciones2[Acciones] Acciones1 --> Join(()) Acciones2 --> Join Join --> Exit(()) </pre>	SI (condición) ENTONCES //Acciones si es verdadera la condición SINO //Acciones si es falsa la condición FIN DEL SI	if (condición) { Acciones a realizar si se cumple la condición } else { Acciones a realizar si no se cumple la condición }
Condicional múltiple	<pre> graph TD Cond{{Condición}} -- Caso 1 --> Acciones1[Acciones 1] Cond -- Caso 2 --> Acciones2[Acciones 2] Cond -- Caso 3 --> Acciones3[Acciones 3] Acciones1 --> Join(()) Acciones2 --> Join Acciones3 --> Join Join --> Exit(()) </pre>	SELECCIONAR CASO CASO 1 //Acciones FIN DEL CASO 1 CASO 2 //Acciones CASO N //Acciones FIN CASO N DEFAULT //Acciones FIN SELECCIONAR	switch (condición) { case 1: //Acciones break; case 2: //Acciones break; case N: //Acciones break; default: //Acciones }

--	--	--	--

2.- Realiza un programa que permita seleccionar entre 3 programas

a) Programa que ejecute una condición que siempre sea verdadera e imprima "Fundamentos de programación"(Utiliza valores booleanos)

b) Programa que ejecute una condición compuesta y que siempre sea falsa e imprima en el else "Error"(Utiliza valores booleanos)

c) Programa del condicional anidado con condicional ternario y comentarios en el código fuente (//Comentar las líneas modificadas)

```

9  #include <stdio.h>
10
11 int main()
12 {
13     char op = '\0';
14     printf("Selecciona una opcion\n");
15     printf("a) Fundamentos \n");
16     printf("b) Falsa\n");
17     printf("c) Ternario\n");
18     scanf("%c",&op);
19     switch(op)
20     {
21         case 'a':
22             if(1)
23             {
24                 printf("Esta instruccion nunca se ejecuta\n");
25             }
26             break;
27         case 'b':
28             if (0)
29             {
30                 printf("Esta instruccion nunca se ejecuta\n");
31             }
32             else
33             {
34                 printf("ERROR\n");
35             }
36             break;
37         case 'c':
38             int uno, dos, tres;
39             printf ("Ingresa 3 numeros separados por espacios:\n");
40             scanf ("%d %d %d", &uno, &dos, &tres);
41             if (uno > dos)
42             {
43                 if(dos > tres)
44                 {
45                     printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres);
46                 }
47                 else
48                 {
49                     (uno > tres)? uno : tres; //cambio a Ternario
50                 }
51             }
52             else
53             {
54                 if (dos > tres)
55                 {
56                     (tres > uno)? tres: uno; //segundo cambio a ternario
57                 }
58                 else
59                 {
60                     printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
61                 }
62                 break;
63             }
64             return (0);
65     }

```

3.- Del programa del condicional anidado, sustituir los if-else por el condicional ternario (?), según corresponda y agregar la condición para valores iguales.

```

#include <stdio.h>
int main()
{
    int uno, dos, tres;
    printf ("Ingrese 3 números separados por espacios:\n");
    scanf ("%d %d %d", &uno, &dos, &tres);
    if (uno > dos)
    {
        if (dos > tres)
        {
            printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres);
        }
        else
        {
            if (uno > tres)
            {
                printf("%d es mayor a %d que es mayor a %d\n", uno, tres, dos);
            }
            else
            {
                printf("%d es mayor a %d que es mayor a %d\n", tres, uno, dos);
            }
        }
    }
    else
    {
        if (dos > tres)
        {
            if (tres > uno)
            {
                printf("%d es mayor a %d que es mayor a %d\n", dos, tres, uno);
            }
            else
            {
                printf("%d es mayor a %d que es mayor a %d\n", dos, uno, tres);
            }
        }
        else
        {
            printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
        }
    }

    return 0;
}

```

```

#include <stdio.h>
int main()
{
    int uno, dos, tres;
    printf ("Ingrese 3 números separados por espacios:\n");
    scanf ("%d %d %d", &uno, &dos, &tres);
    max=(uno > dos)?
        max=(dos > tres)? printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres):
        max=(uno > tres)? printf("%d es mayor a %d que es mayor a %d\n", uno, tres, dos):
        printf("%d es mayor a %d que es mayor a %d\n", tres, uno, dos):
    max=(dos > tres)?
        max=(tres > uno)?printf("%d es mayor a %d que es mayor a %d\n", dos, tres, uno):
        printf("%d es mayor a %d que es mayor a %d\n", dos, uno, tres):
    printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
}

```

Así quedaría una vez sustituido el if, por max. A la hora de compilar me marca un error, que es el siguiente:

```

main.c: In function 'main':
main.c:15:2: error: 'max' undeclared (first use in this function)
   15 |   max=(uno > dos)?
      |   ^~~~
main.c:15:2: note: each undeclared identifier is reported only once for each function it appears in

```

La verdad no entendí muy bien lo que tenía que corregir, por eso ya no seguí más.

Ojalá pueda explicarme lo que tengo que hacer en este caso.

4.- Del programa del condicional ternario(?), sustituir por if-else, según corresponda y agregar la condición para valores iguales.

Programa9.c

```
#include <stdio.h>
int main()
{
    double a, b, res;
    printf("Calcular el error matemático E = |a - b|\n\n");
    printf("Ingrese el valor de a:\n");
    scanf("%lf",&a);
    printf("Ingrese el valor de b:\n");
    scanf("%lf",&b);
    res = a < b ? b-a : a-b;
    printf("El error matemático de\n");
    printf("| %lf - %lf | es %lf\n", a, b, res);
    return 0;
}
```

```
9  #include <stdio.h>
10
11  int main()
12  {
13      double a, b, res;
14      printf("Calcular el error matemático E = |a - b|\n\n");
15      printf("Ingrese el valor de a:\n");
16      scanf("%lf",&a);
17      printf("Ingrese el valor de b:\n");
18      scanf("%lf",&b);
19      if(a<b)
20      {
21          printf("El error matemático de\n");
22          printf("| %lf - %lf | es %lf\n", a, b, res);
23      }
24      else
25          printf("El error matemático de\n");
26          printf("| %lf - %lf | es %lf\n", b, a, res);
27  return 0;
28  }
29
```

input

```
Ingrese el valor de a:
8
Ingrese el valor de b:
6
El error matemático de
| 6.000000 - 8.000000 | es 0.000000

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusiones

Gracias a esta práctica me puedo dar cuenta sobre importancia de las estructuras como la condicional o la selectiva, ya que gracias a estas podemos realizar programas, además esta es una de las más sencillas. Podemos hacer diferentes programas que contengan menús.

Se me complicó más que nada los ejercicios de tarea debido a que en uno teníamos que hacer un menú con ciertas restricciones o ciertos pasos.

Referencias

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- <http://lcp02.fi-b.unam.mx/>