	Práctica
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>	Grupo 3 Bloque 135
<i>No de Práctica(s):</i>	Practica #8
<i>Integrante(s):</i>	Muñoz Reyes Laura Vanessa Gutiérrez Ramos Alitzel Anaid
<i>No. de Equipo de cómputo empleado:</i>	Luxemburgo 10
<i>No. de Lista o Brigada:</i>	No. de Lista Muñoz 33 No. de Lista Gutiérrez 21
<i>Semestre:</i>	2020-01
<i>Fecha de entrega:</i>	07 de octubre del 2019
<i>Observaciones:</i>	Les dije que pusieran los números de cuanta de ambas Si esto vuelve a suceder, solo evaluaré a la que subió la práctica.

CALIFICACIÓN: 10

Practica #8.

Estructuras de Selección

INTRODUCCION:

Existen programas donde podemos obtener diferentes resultados, de acuerdo a los tipos de datos a ejecutar o las funciones que busquemos realizar. En programación C, los dos tipos de estructura de selección son sentencias "if" y casos de "switch". Es importante que conozcamos este tipo de estructuras para realizar diferentes procesos en programas de nivel básico y de un nivel más complejo.

OBJETIVO:

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria para la resolución de problemas básicos.

MARCO REFERENCIAL:

En programación, la estructura de selección es un tipo de estructura de control. También llamada estructura de decisión o estructura selectiva.

En una estructura de selección, el algoritmo al ser ejecutado toma una decisión: ejecutar o no ciertas instrucciones, si se cumplen o no ciertas condiciones. Las condiciones devuelven un valor, verdadero o falso, determinado así la secuencia a seguir.

Existen tres tipos de estructuras de selección:

IF

Una declaración "if" te permite cambiar el flujo de tu programa. Puedes comparar una variable con un valor o varios valores, usando bloques de "else if", y decidir qué haces si la variable no coincide con ninguno de ellos, y qué hacer cuando no es así, mediante el uso de un bloque de "else".

Puedes ampliar la funcionalidad de una sentencia "if" comparando más de una variable y usando a la vez la lógica "and" y "or", o puedes anidar varias declaraciones "if" dentro de sí para comparar varias variables.

Sintaxis de la declaración "if"

Para usar una declaración "if", tienes que escribir "if (var op value)" donde "var" es la variable que quieres comparar, "op" es el operador, ya sea igual, mayor o menor que, y "value" que es el valor para ver si la variable se evalúa.

Luego del bloque de "if", puedes incluir uno o más bloques de "else if" para probar más condiciones, usando la misma sintaxis de la declaración inicial. También puedes usar "else" como bloque final que se ejecuta si la variable no coincide con ninguno de los bloques de "if" o de "else".

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

SWITCH

La selección de conmutador de casos funciona mejor cuando se tiene varios resultados posibles, aunque se puede utilizar para tan sólo uno. Pasas una variable a la estructura de conmutación y comparas su valor con un número de casos, y se lleva el apropiado de la que coincide. A diferencia de la declaración "if", el caso del conmutador de casos sólo ejecuta los resultados cuando la variable coincide con uno de los casos. No es compatible con mayores o menores que los operadores.

Para utilizar la estructura de la selección de conmutador de casos, escribe "switch (var)" para abrir el bloque, donde "var" es la variable que deseas comparar. En las siguientes líneas, se programan los distintos casos escribiendo "case" y el patrón que desees a la altura, y luego las declaraciones que desees ejecutar para cada caso. Por último, escribe la palabra clave "break" para decirle al bloque que finalice el conmutador cuando el caso termine. Después de cada caso, crea un conmutador de "default", con el que el bloque se ejecuta cuando ninguno de los casos coincide.

```
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*/  
}
```

Operadores Condicionales

Las condiciones que van dentro de los paréntesis usan operadores condicionales, que son los siguientes: <, >, <=, >=, ==, !=. (Menor, mayor, menor-igual, mayor-igual, igual, diferente)

Condición ? SiSeCumple : SiNoSeCumple ;

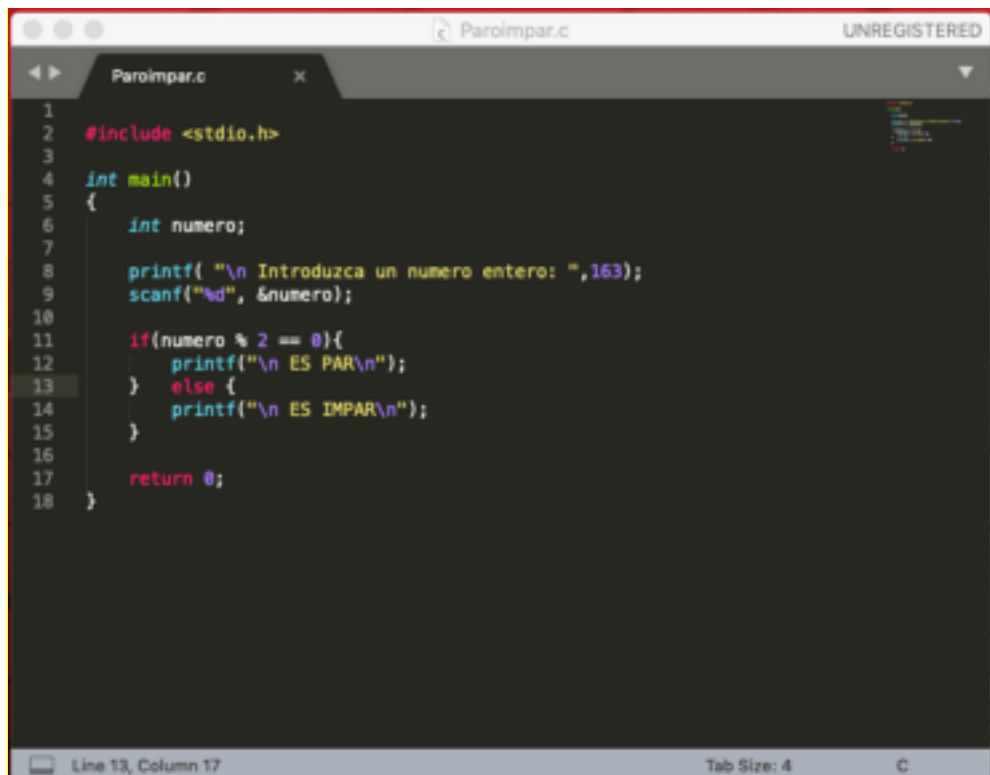
```
int a = 3;
int b = 4;

mayor = a > b ? a : b ;
```

PROCEDIMIENTO / RESULTADO:

1. Experimentar al utilizar diferentes números dentro del condicional del "if" (asegurar usar el 0 entre las elecciones).
Hacer un programa que lea un número e indique si es par o non:

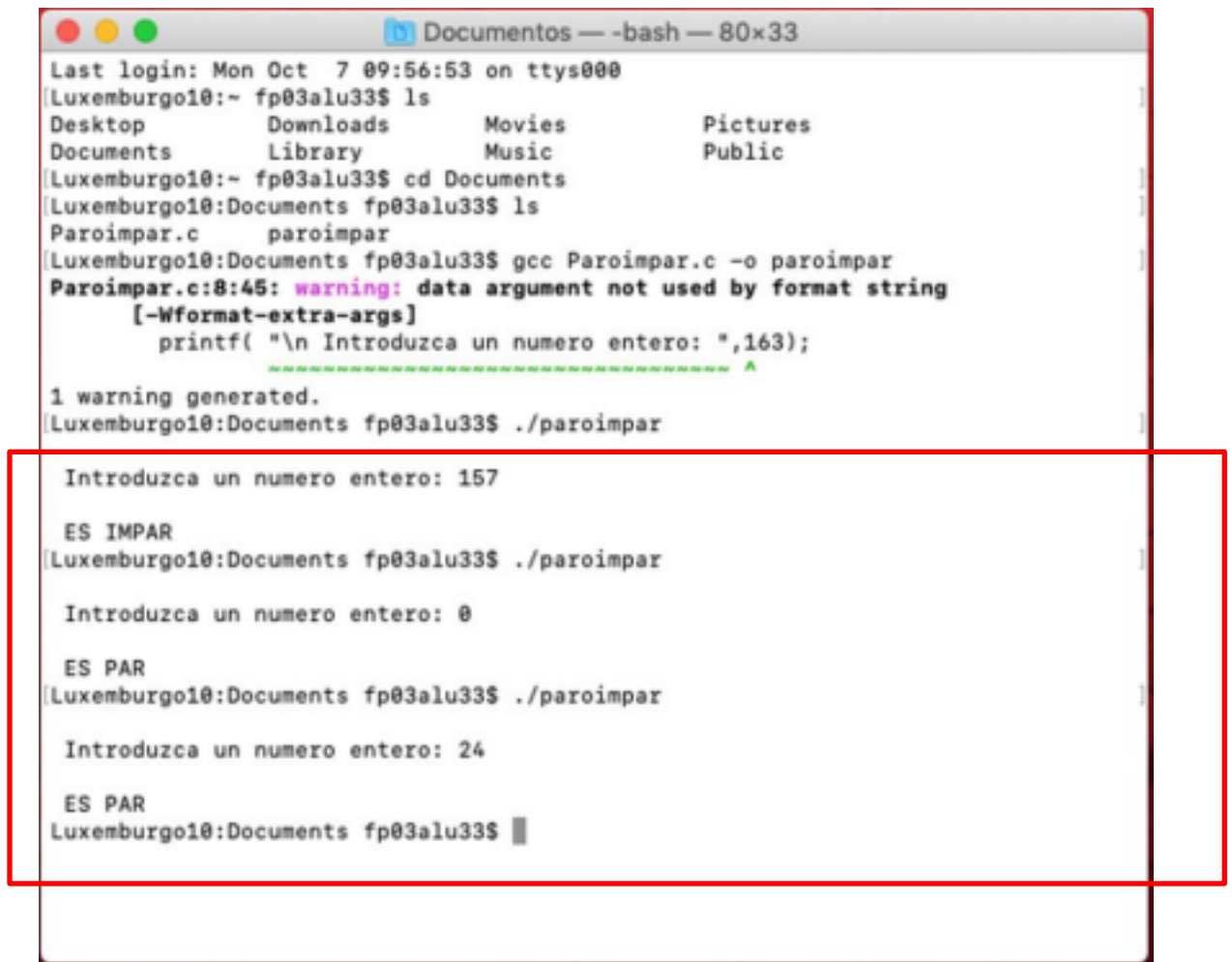
→ Programa desarrollado en sublimetext



```
1
2 #include <stdio.h>
3
4 int main()
5 {
6     int numero;
7
8     printf( "\n Introduzca un numero entero: ",163);
9     scanf("%d", &numero);
10
11     if(numero % 2 == 0){
12         printf("\n ES PAR\n");
13     } else {
14         printf("\n ES IMPAR\n");
15     }
16
17     return 0;
18 }
```

Line 13, Column 17 Tab Size: 4 C

→ Programa Compilado y Probado con 157, 0 , 24



```
Documents — -bash — 80x33
Last login: Mon Oct 7 09:56:53 on ttys000
Luxemburgo10:~ fp03alu33$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
Luxemburgo10:~ fp03alu33$ cd Documents
Luxemburgo10:Documents fp03alu33$ ls
Paroimpar.c  paroimpar
Luxemburgo10:Documents fp03alu33$ gcc Paroimpar.c -o paroimpar
Paroimpar.c:8:45: warning: data argument not used by format string
[-Wformat-extra-args]
    printf( "\n Introduzca un numero entero: ",163);
    ^
1 warning generated.
Luxemburgo10:Documents fp03alu33$ ./paroimpar

Introduzca un numero entero: 157

ES IMPAR
Luxemburgo10:Documents fp03alu33$ ./paroimpar

Introduzca un numero entero: 0

ES PAR
Luxemburgo10:Documents fp03alu33$ ./paroimpar

Introduzca un numero entero: 24

ES PAR
Luxemburgo10:Documents fp03alu33$
```

2. Programa que lea una letra e indique si es vocal o consonante:

→ Programa elaborado en Sublimetext, al momento de elaborarlo nos percatamos de que también debían incluirse las mayúsculas pero al ser switch lo mejor era ponerlo en casos distintos.

Vocal.CUNREGISTERED

Paroimpar.cVocal.C

```
2  #include <stdio.h>
3
4  int main ()
5  {
6
7      char letra;
8
9      printf( "\n Introduzca una letra: " );
10     scanf( "%c" , &letra);
11
12     switch (letra){
13         case 'a':
14             printf( "\n ES UNA VOCAL\n");
15             break;
16
17         case 'A':
18             printf( "\n ES UNA VOCAL\n");
19             break;
20
21         case 'E':
22             printf( "\n ES UNA VOCAL\n");
23             break;
24
25         case 'e':
26             printf( "\n ES UNA VOCAL\n");
27             break;
28
29         case 'I':
30             printf( "\n ES UNA VOCAL\n");
31             break;
32
33         case 'i':
34             printf( "\n ES UNA VOCAL\n");
35             break;
36
37         case 'O':
38             printf( "\n ES UNA VOCAL\n");
39             break;
40
41         case 'o':
42             printf( "\n ES UNA VOCAL\n");
43             break;
44
45         case 'U':
46             printf( "\n ES UNA VOCAL\n");
47             break;
48
49         case 'u':
50             printf( "\n ES UNA VOCAL\n");
51             break;
52
53         default:
54             printf( "\n ES UNA CONSONANTE\n");
55
56     }
57
58     return 0;
59
60 }
61
```

Line 52, Column 1Tab Size: 4C++

→ Programa compilado y probado con e, l, m, S

```
Documents — -bash — 80x35
Last login: Mon Oct 7 10:37:50 on ttys000
Luxemburgo10:~ fp03alu33$ cd Documents
Luxemburgo10:Documents fp03alu33$ ls
Paroimpar.c  Vocal.C      paroimpar     vocal
Luxemburgo10:Documents fp03alu33$ gcc Vocal.c -o vocal
Luxemburgo10:Documents fp03alu33$ ./vocal

  Introduzca una letra: e

  ES UNA VOCAL
Luxemburgo10:Documents fp03alu33$ ./vocal

  Introduzca una letra: I

  ES UNA VOCAL
Luxemburgo10:Documents fp03alu33$ ./vocal

  Introduzca una letra: m

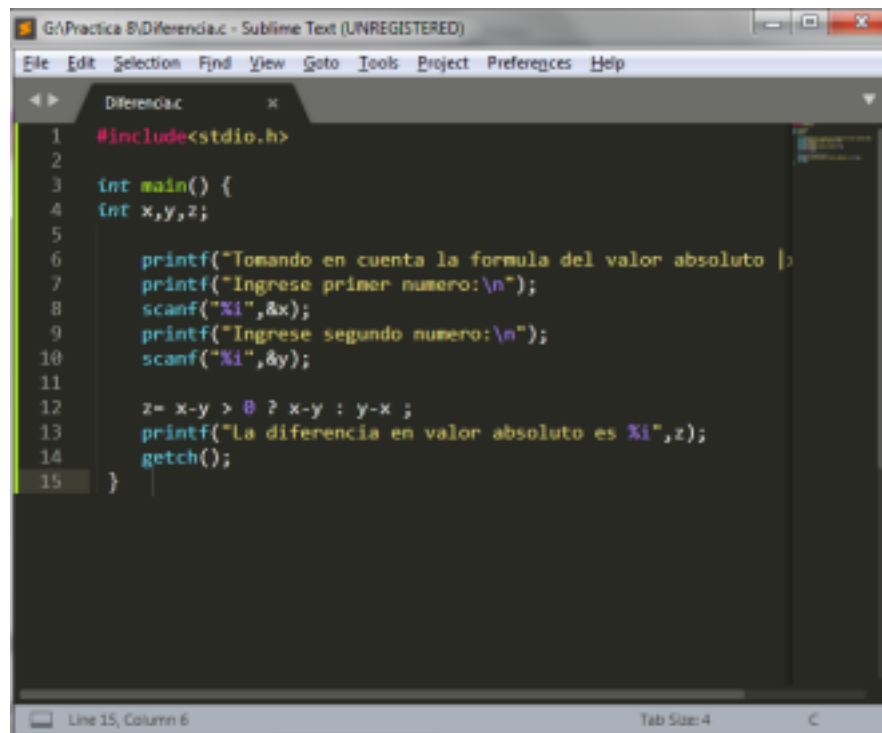
  ES UNA CONSONANTE
Luxemburgo10:Documents fp03alu33$ ./vocal

  Introduzca una letra: S

  ES UNA CONSONANTE
Luxemburgo10:Documents fp03alu33$
```

3. Use condicional para hacer un programa que obtenga el valor absoluto de la diferencia entre dos números:

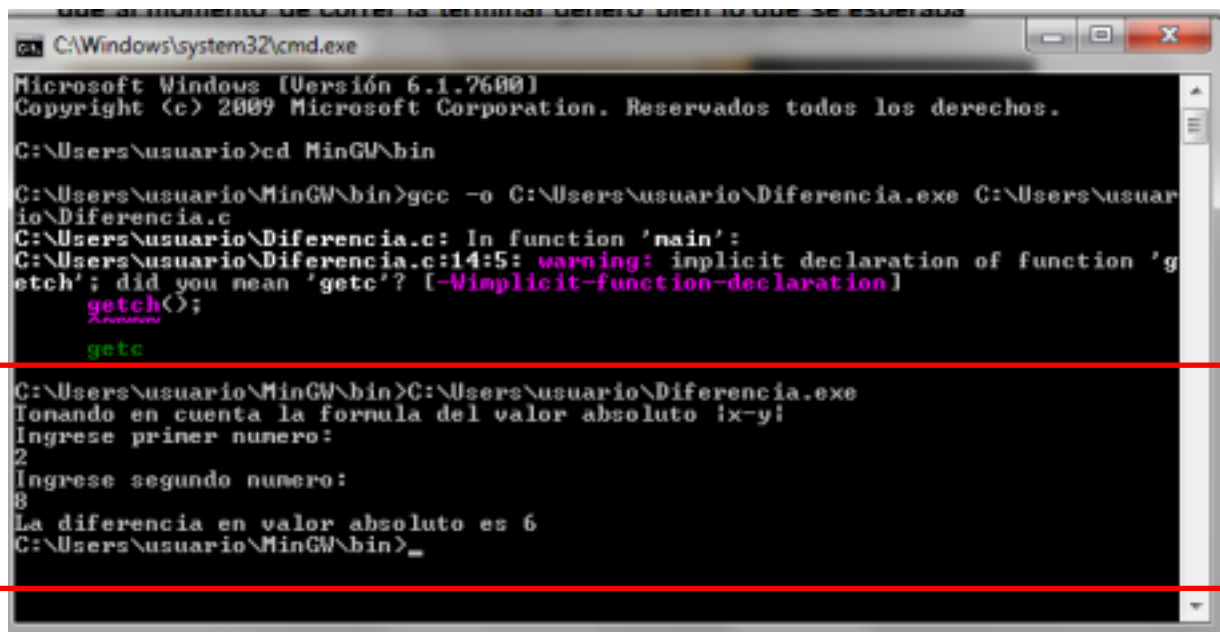
→ Programa desarrollado en Sublimetext



```
GAPractica 8/Diferencia.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Diferencia.c
1 #include<stdio.h>
2
3 int main() {
4     int x,y,z;
5
6     printf("Tomando en cuenta la formula del valor absoluto |>
7     printf("Ingrese primer numero:\n");
8     scanf("%i",&x);
9     printf("Ingrese segundo numero:\n");
10    scanf("%i",&y);
11
12    z= x-y > 0 ? x-y : y-x ;
13    printf("La diferencia en valor absoluto es %i",z);
14    getch();
15 }
```

→ Se compilo y probó con (2 – 8). Nota: se compilo en Windows usando el programa alterno MinGW y teniendo un archivo de salida con terminación .exe



```
cmd C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\usuario>cd MinGW\bin

C:\Users\usuario\MinGW\bin>gcc -o C:\Users\usuario\Diferencia.exe C:\Users\usuar
io\Diferencia.c
C:\Users\usuario\Diferencia.c: In function 'main':
C:\Users\usuario\Diferencia.c:14:5: warning: implicit declaration of function 'g
etch'; did you mean 'getc'? [-Wimplicit-function-declaration]
    getch();
    ^~~~~~
    getc

C:\Users\usuario\MinGW\bin>C:\Users\usuario\Diferencia.exe
Tomando en cuenta la formula del valor absoluto |x-y|
Ingrese primer numero:
2
Ingrese segundo numero:
8
La diferencia en valor absoluto es 6
C:\Users\usuario\MinGW\bin>_
```


CONCLUSIONES

Como conclusión es importante hacer un uso correcto de las estructuras de selección, a la hora de programar en C. Ya que por ejemplo al realizar el programa con switch estábamos cometiendo el mismo error de poner operaciones dentro del bloque de case y nos salía error. Si sabes bien cual condicional usar en cada tipo de problema es más fácil usar las herramientas adecuadas para tu problema.

Fuentes

<https://drive.google.com/drive/folders/1IN9FAXGrvirYqKh1HV-wUaNLZJ0MZwdJ>

https://techlandia.com/estructura-seleccion-programacion-info_504218/

https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Instrucciones_de_control

<https://dcodingames.com/estructuras-de-seleccion/>