



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: MC. Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): Práctica 8

Integrante(s): Martínez Marcelino Dalila

*No. de Equipo de
cómputo empleado:* No. de cuenta: 313080119

No. de Lista o Brigada:

Semestre: 2020-1

Fecha de entrega: Lunes 7 de octubre de 2019

Observaciones: Excelente

CALIFICACIÓN: 10

Práctica No. 8

Introducción:

En esta práctica el estudiante creará programas en Lenguaje C para resolver los problemas planteados. Utilizará estructuras condicionales como son la estructura if o if-else, en donde el programa requiera expresar alguna decisión de verdadero o falso, también utilizará la estructura que lleva por nombre condicional, que cumple con la misma función que la estructura if, y, por último, la estructura switch case la cual se utilizará en donde se requiera tomar una decisión sobre un valor específico.

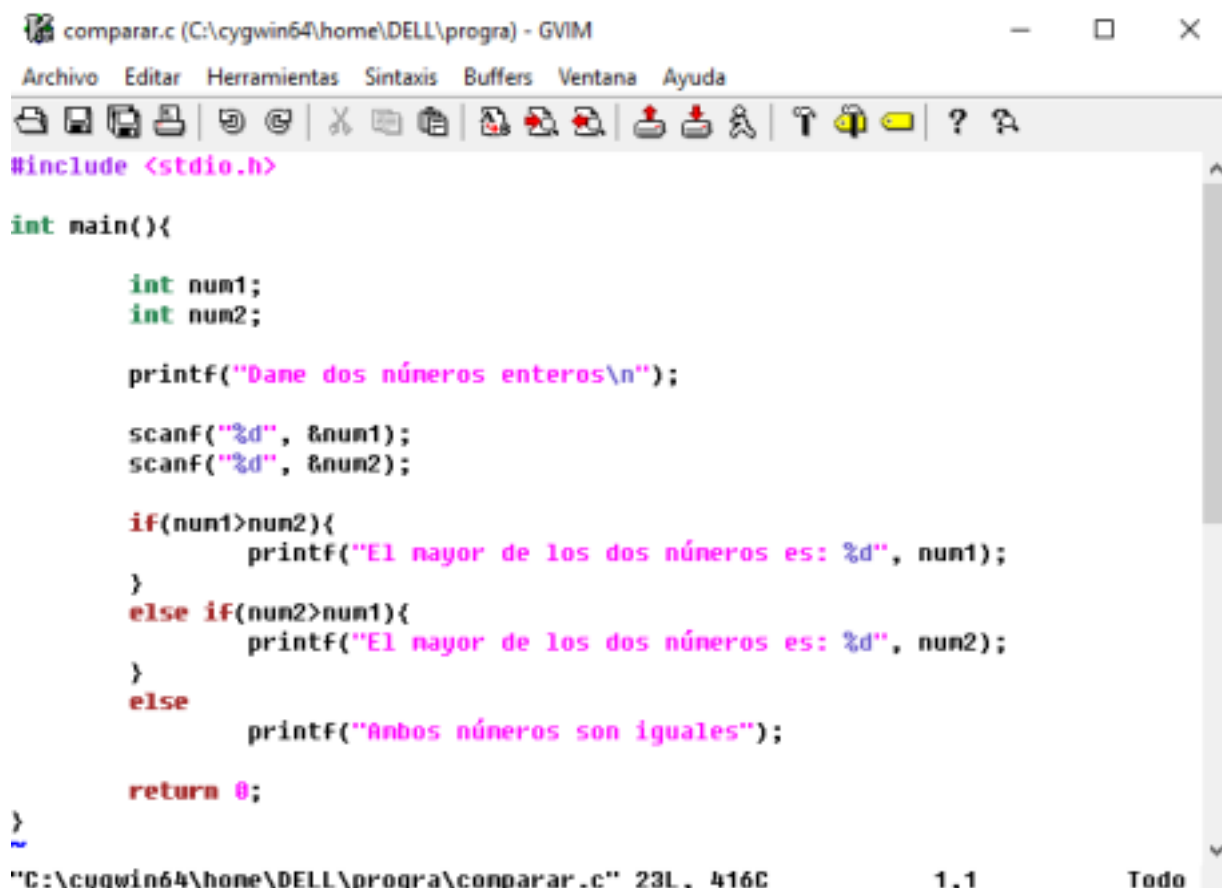
Objetivo:

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

Actividad 1.

Experimentar al utilizar diferentes números dentro del condicional del "if" (asegúrense de usar el 0 entre sus elecciones).

✚ El siguiente programa indica el mayor de dos números o si ambos números son iguales.



```
comparar.c (C:\cygwin64\home\DELL\progra) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda
[Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons]
#include <stdio.h>

int main(){

    int num1;
    int num2;

    printf("Dane dos números enteros\n");

    scanf("%d", &num1);
    scanf("%d", &num2);

    if(num1>num2){
        printf("El mayor de los dos números es: %d", num1);
    }
    else if(num2>num1){
        printf("El mayor de los dos números es: %d", num2);
    }
    else
        printf("Ambos números son iguales");

    return 0;
}

"C:\cygwin64\home\DELL\progra\comparar.c" 23L, 416C 1,1 Todo
```

Y al compilarlo veremos lo que hace el programa, al darle dos números diferentes.

```
DELL@DESKTOP-NJV5QTU ~/progra
$ ./comparar
Dame dos números enteros
0
5
El mayor de los dos números es: 5
```

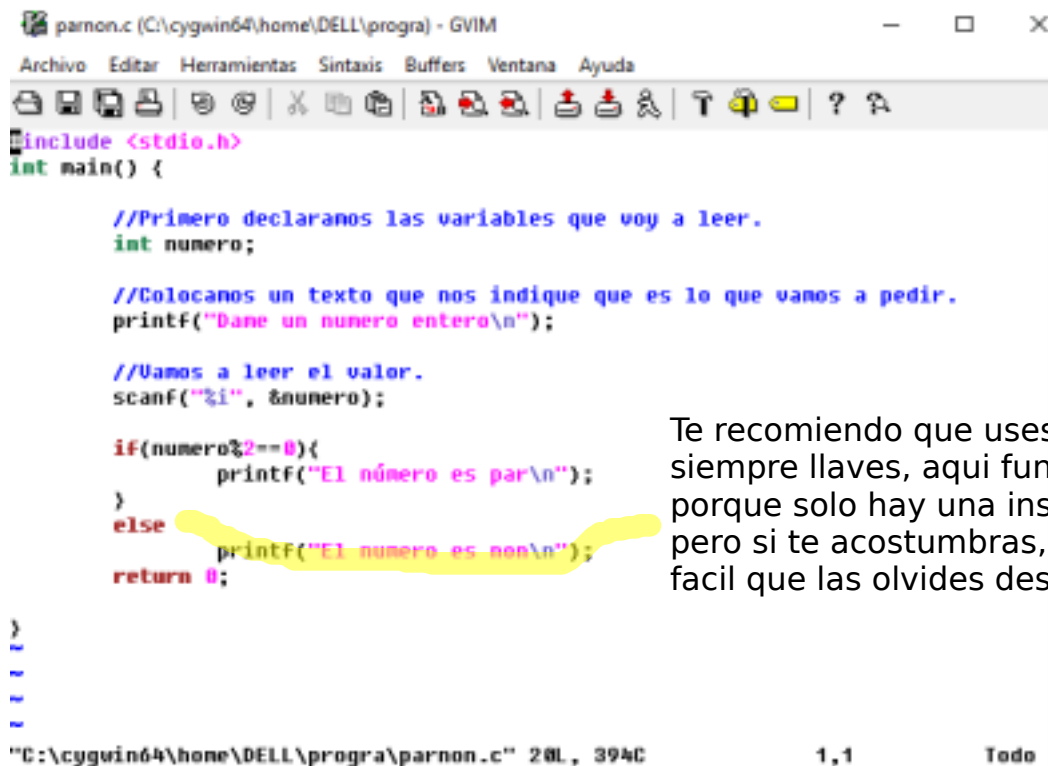
Si le diéramos dos números iguales entonces, sería lo siguiente:

```
DELL@DESKTOP-NJV5QTU ~/progra
$ ./comparar
Dame dos números enteros
12
12
Ambos números son iguales
```

Actividad 2.

Hacer un programa que lea un número e indique si es par o non.

Para este programa ocuparemos la estructura de if-else.



```
parnon.c (C:\cygwin64\home\DELL\progra) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda
[Icons]
#include <stdio.h>
int main() {

    //Primero declaranos las variables que voy a leer.
    int numero;

    //Colocamos un texto que nos indique que es lo que vamos a pedir.
    printf("Dame un numero entero\n");

    //Vamos a leer el valor.
    scanf("%i", &numero);

    if(numero%2==0){
        printf("El número es par\n");
    }
    else
        printf("El numero es non\n");
    return 0;

}
~
~
~
"C:\cygwin64\home\DELL\progra\parnon.c" 28L, 394C      1,1      Todo
```

Te recomiendo que uses siempre llaves, aqui funciona porque solo hay una instrucción, pero si te acostumbras, es facil que las olvides después

Y al compilarlo vemos que el programa indicara si el número dado es par o non. Para ello veremos tres pruebas.

Primero con un número non.

```
DELL@DESKTOP-NJV5QTU ~/progra
$ gcc parnon.c -o parnon

DELL@DESKTOP-NJV5QTU ~/progra
$ ./parnon
Dame un numero entero
7
El numero es non
```

El siguiente será con un número que es par.

```
DELL@DESKTOP-NJVSQTU ~/progra
$ ./parnon
Dame un numero entero
10
El número es par
```

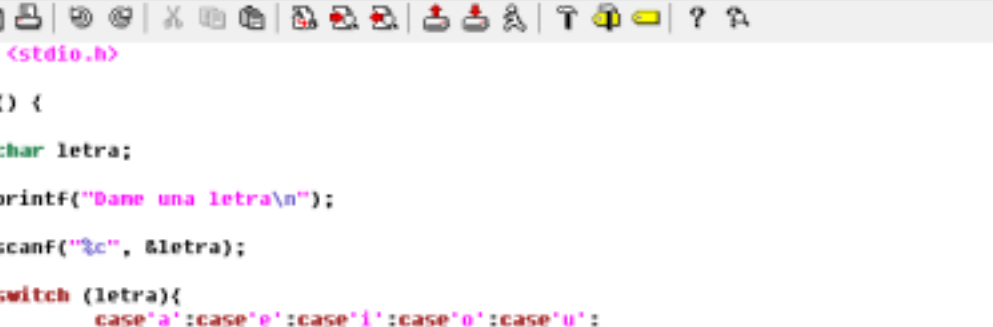
Y por último probaremos con el numero cero. El cual resulta que es par.

```
DELL@DESKTOP-NJVSQTU ~/progra
$ ./parnon
Dame un numero entero
0
El número es par
```

Actividad 3.

Hacer un programa que lea una letra e indique si es vocal o consonante.

En este ocuparemos la estructura de switch case.



The screenshot shows a Gvim editor window titled "conso vocal.c (C:\cygwin64\home\DELL\progra) - GVIM". The menu bar includes "Archivo", "Editar", "Herramientas", "Sintaxis", "Buffers", "Ventana", and "Ayuda". The toolbar contains various icons for file operations and editing. The code in the editor is as follows:

```
#include <stdio.h>

int main() {

    char letra;

    printf("Dane una letra\n");

    scanf("%c", &letra);

    switch (letra){
        case 'a':case 'e':case 'i':case 'o':case 'u':
            printf("%c\t", letra);
            printf("Es una vocal");
            break;
        case 'b':case 'c':case 'd':case 'f':case 'g':case 'h':case 'j':case 'k':case 'l':case 'm':
        case 'n':case 'p':case 'q':case 'r':case 's':case 't':case 'v':case 'w':case 'x':case 'y':case 'z':
            printf("%c\t", letra);
            printf("Es una consonante");
            break;
        default:
            printf("Esto no es una letra");
            break;
    }

    return 0;
}
```

The status bar at the bottom right shows "24.3-17" and "Todo".

Quando lo compilamos el programa indica si la letra dada es una vocal o consonante.

Primero probamos con una consonante.

```
DELL@DESKTOP-NJV5QTU ~/progra
$ gcc consovocal.c -o convo

DELL@DESKTOP-NJV5QTU ~/progra
$ ./convo
Dame una letra
h
h          Es una consonante
```

Ahora probaremos con una vocal.

```
DELL@DESKTOP-NJV5QTU ~/progra
$ ./convo
Dame una letra
o
o      Es una vocal
```

El programa solo reconoce letras, si ingresamos al programa algo diferente de una letra, dará como resultado lo siguiente:

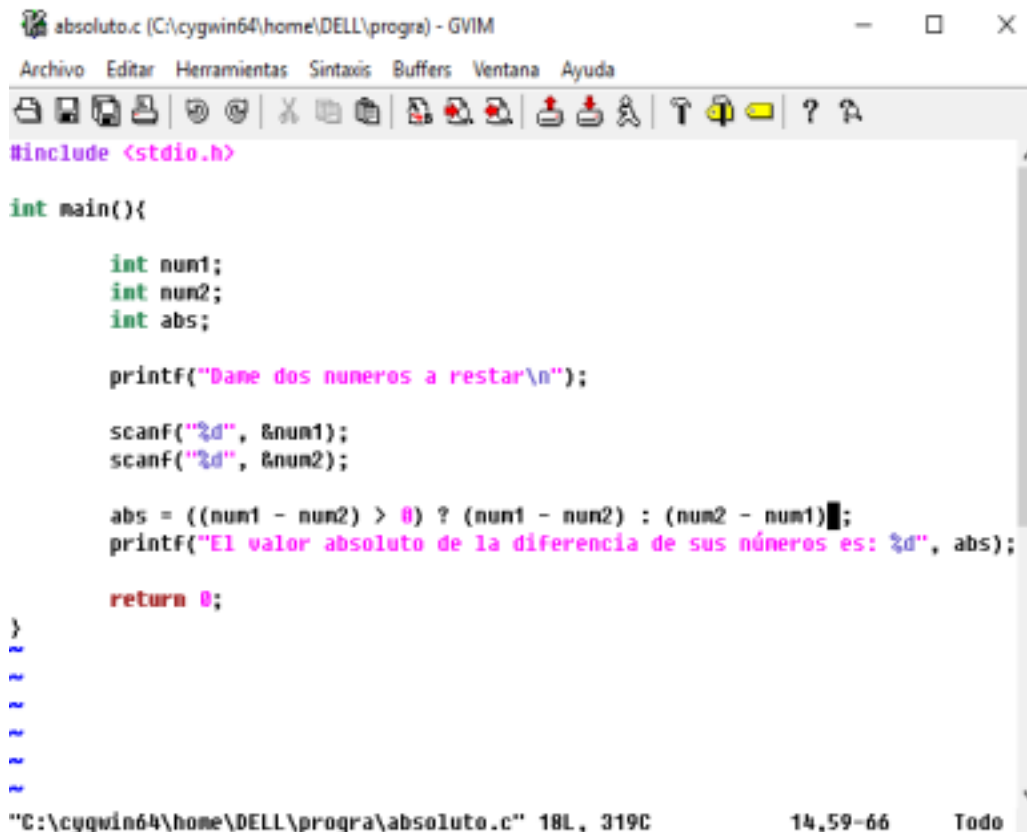
```
DELL@DESKTOP-NJV5QTU ~/progra
$ ./convo
Dame una letra
8
Esto no es una letra
```

El defecto de este programa es que solo acepta letras en minúsculas, si se ingresa una letra en mayúscula el programa no lo detecta como tal, y entonces, hará lo siguiente:

```
DELL@DESKTOP-NJV5QTU ~/progra
$ ./convo
Dame una letra
A
Esto no es una letra
```

Actividad 4.

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

A screenshot of a text editor window titled 'absoluto.c (C:\cygwin64\home\DELL\progra) - GVIM'. The editor shows a C program that calculates the absolute difference between two integers. The code includes a header file, declares variables, prompts the user for two numbers, reads them, and then uses a ternary operator to calculate the absolute difference. The status bar at the bottom shows the file path, line 18, column 19, and a total of 14,59-66 lines.

```
absoluto.c (C:\cygwin64\home\DELL\progra) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

#include <stdio.h>

int main(){

    int num1;
    int num2;
    int abs;

    printf("Dame dos numeros a restar\n");

    scanf("%d", &num1);
    scanf("%d", &num2);

    abs = ((num1 - num2) > 0) ? (num1 - num2) : (num2 - num1);
    printf("El valor absoluto de la diferencia de sus números es: %d", abs);

    return 0;
}

~
~
~
~
~

"C:\cygwin64\home\DELL\progra\absoluto.c" 18L, 319C    14,59-66    Todo
```

Al compilar el programa el resultado que arrojará será siempre positivo, dado que es valor absoluto.

Vemos lo que pasa si le damos un número negativo y un cero.

```
DELL@DESKTOP-NJVSQTU ~/progra
$ gcc absoluto.c -o abso

DELL@DESKTOP-NJVSQTU ~/progra
$ ./abso
Dame dos numeros a restar
-15
0
El valor absoluto de la diferencia de sus números es: 15
```

Si le damos un número que sea menor al siguiente, aun así dará positivo.

```
DELL@DESKTOP-NJVSQTU ~/progra
$ ./abso
Dame dos numeros a restar
15
20
El valor absoluto de la diferencia de sus números es: 5
```

Si hacemos una resta en donde ambos números son negativos, se verá lo siguiente.

```
DELL@DESKTOP-NJVSQTU ~/progra
$ ./abso
Dame dos numeros a restar
-13
-10
El valor absoluto de la diferencia de sus números es: 3
```

Conclusión.

El hacer programas en lenguaje C no resulta complicado, solo tenemos que aprender muy bien los elementos que lleva su sintaxis, pues el hecho de que llegues a escribir algo mal, el programa no lo ejecutará, un ejemplo de lo que digo es que al equivocarme y escribir `prinft` en lugar de `printf` el programa no se ejecutó, a pesar que todo lo demás estuviera bien estructurado.

También debemos de especificar con qué tipo de variable estamos trabajando, así como donde debemos de colocar las llaves para abrir o cerrar un bloque. Y por último esta práctica nos sirvió para conocer algunos de los diferentes tipos de estructura y que es lo que se ejecuta en ellos, pues para algunos problemas convendrá utilizar ciertos tipos de estructura, y esto dependerá de lo que queramos que haga nuestro programa.