|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodriguez Espino |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 4 |
| *No de Práctica(s):* | 8 |
| *Integrante(s):* | Garduño Pérez Angel Isaac |
|  |  |
| *No. de Equipo de cómputo empleado* | 16 |
| *Semestre:* | 2 |
| *Fecha de entrega:* | 05/04/2019 |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Práctica 08: Estructuras de selección

**Objetivo:**

Elaborar 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**

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el

procesamiento de datos.

Las estructuras de selección (o condicionales) permiten realizar una u otra acción con base

en una expresión lógica. Las acciones posibles a realizar son mutuamente excluyentes, es

decir, solo se puede ejecutar una a la vez dentro de toda la estructura.

Lenguaje C posee 3 estructuras de selección: la estructura if-else, la estructura switch y la

estructura condicional o ternaria.

**Estructura de control selectiva if**

La estructura de control de flujo más simple es la estructura condicional **if**, su sintaxis es la

siguiente:

**if (expresión\_lógica) {**

**// bloque de código a ejecutar**

**}**

En esta estructura se evalúa la expresión lógica y, si se cumple (si la condición es

verdadera), se ejecutan las instrucciones del bloque que se encuentra entre las llaves de la

estructura. Si no se cumple la condición, se continúa con el flujo normal del programa.

**Estructura de control selectiva if-else**

La sintaxis de la estructura de control de flujo **if-else** es la siguiente:

**if (expresión\_lógica) {**

**// bloque de código a ejecutar**

**// si la condición es verdadera**

**} else {**

**// bloque de código a ejecutar**

**// si la condición es falsa**

**}**

Esta estructura evalúa la expresión lógica y si la condición es verdadera se ejecutan las

instrucciones del bloque que se encuentra entre las primeras llaves, si la condición es falsa

se ejecuta el bloque de código que está entre las llaves después de la palabra reservada

'else'. Al final de que se ejecute uno u otro código, se continúa con el flujo normal del

programa.

Es posible *anidar* varias estructuras if-else, es decir, dentro de una estructura if-else tener

una o varias estructuras if-else.

**Estructura de control selectiva switch-case**

La sintaxis de la estructura switch-case es la siguiente:

**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\*/

**}**

La estructura switch-case evalúa la variable que se encuentra entre paréntesis después de

la palabra reservada switch y la compara con los valores constantes que posee cada caso

(case). Los tipos de datos que puede evaluar esta estructura son enteros, caracteres y

enumeraciones. Al final de cada caso se ejecuta la instrucción break, si se omite esta

palabra reservada se ejecutaría el siguiente caso, es decir, se utiliza para indicar que el

bloque de código a ejecutar ya terminó y poder así salir de la estructura.

Si la opción a evaluar no coincide dentro de algún caso, entonces se ejecuta el bloque por

defecto (default). El bloque por defecto normalmente se escribe al final de la estructura,

pero se puede escribir en cualquier otra parte. Si se escribe en alguna otra parte el bloque

debe terminar con la palabra reservada break.

**Enumeración**

Existe otro tipo de dato constante conocido como enumeración. Una variable enumerador

se puede crear de la siguiente manera:

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

Para crear una enumeración se utiliza la palabra reservada enum, seguida de un

identificador (nombre) y, entre llaves se ingresan los nombres de los valores que puede

tomar dicha enumeración, separando los valores por coma. Los valores son elementos

enteros y constantes (por lo tanto, se escriben con mayúsculas).

**Ejemplo**

**enum boolean {FALSE, TRUE};**

La enumeración se llama 'boolean' y contiene dos elementos, el primero (FALSE) posee el

valor 0 y el siguiente (TRUE) posee el valor 1. Si hubiese más elementos en la

enumeración, la numeración correría de manera ascendente.

Es posible cambiar el valor de un elemento, para ello solo se le asigna el valor deseado:

**enum diasSemana {LUNES, MARTES, MIERCOLES=5, JUEVES, VIERNES};**

**Estructura de control selectiva condicional**

La estructura condicional (también llamado operador ternario) permite realizar una

comparación rápida. Su sintaxis es la siguiente:

**Condición ? SiSeCumple : SiNoSeCumple**

Consta de tres partes, una condición y dos acciones a seguir con base en la expresión

condicional. Si la condición se cumple (es verdadera) se ejecuta la instrucción que se

encuentra después del símbolo ‘?’; si la condición no se cumple (es falsa) se ejecuta la

instrucción que se encuentra después del símbolo.

Actividades:

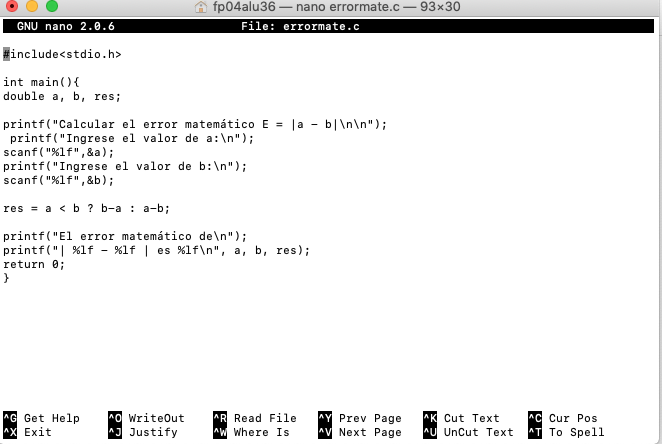
1. Utilizando swithc case realizar un programa que permita elegir una opción del menú a partir del entero ingresado. La opción se lee desde la entrada estándar (el teclado).

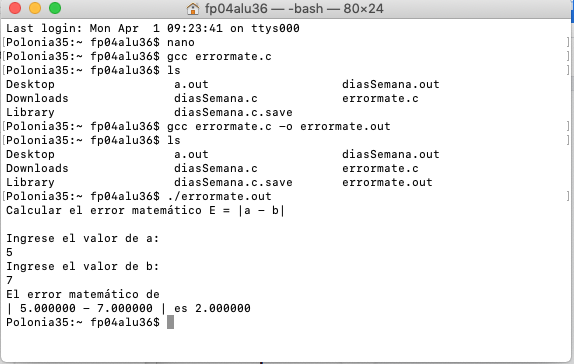


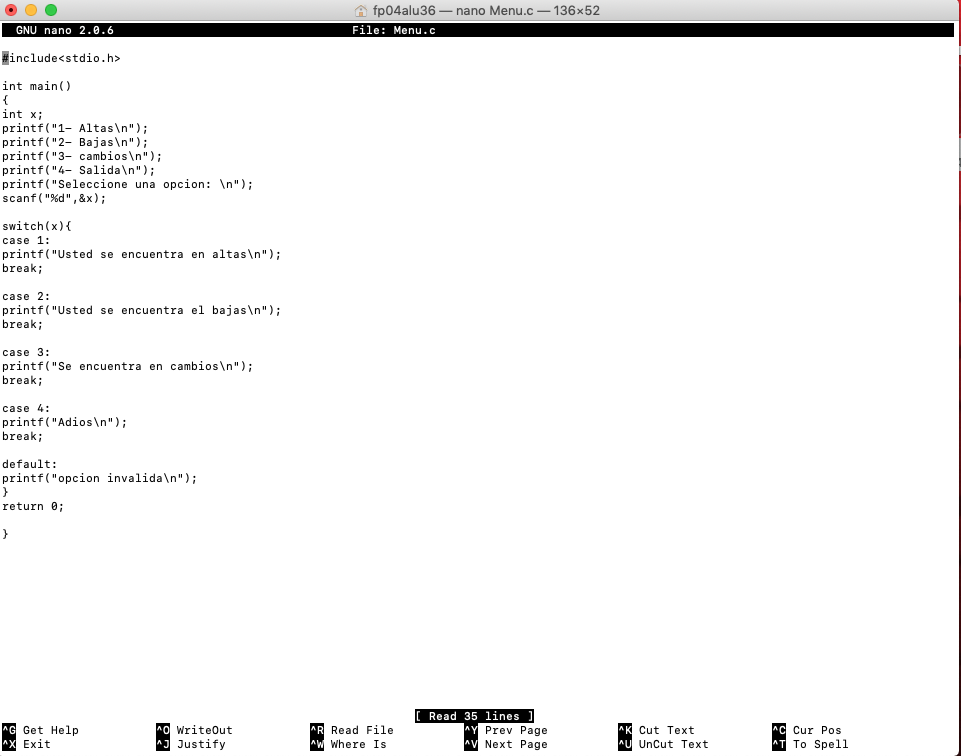


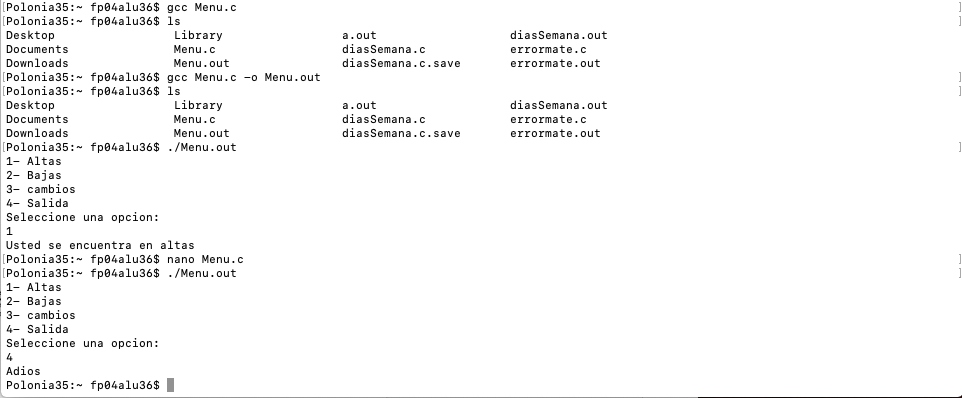
1. Un programa permite calcular el error matemático a partir de dos

valores (a y b) ingresados desde la entrada estándar (el teclado), a partir de la fórmula.

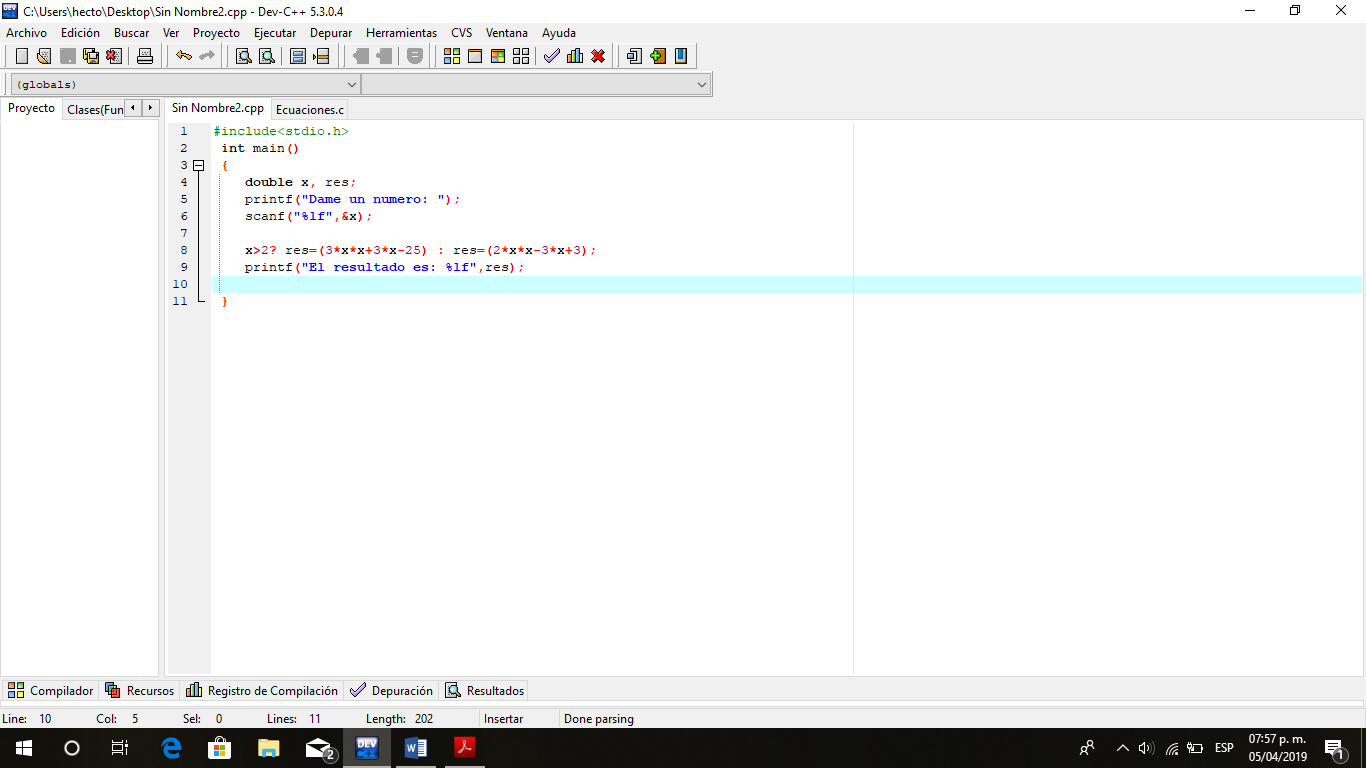


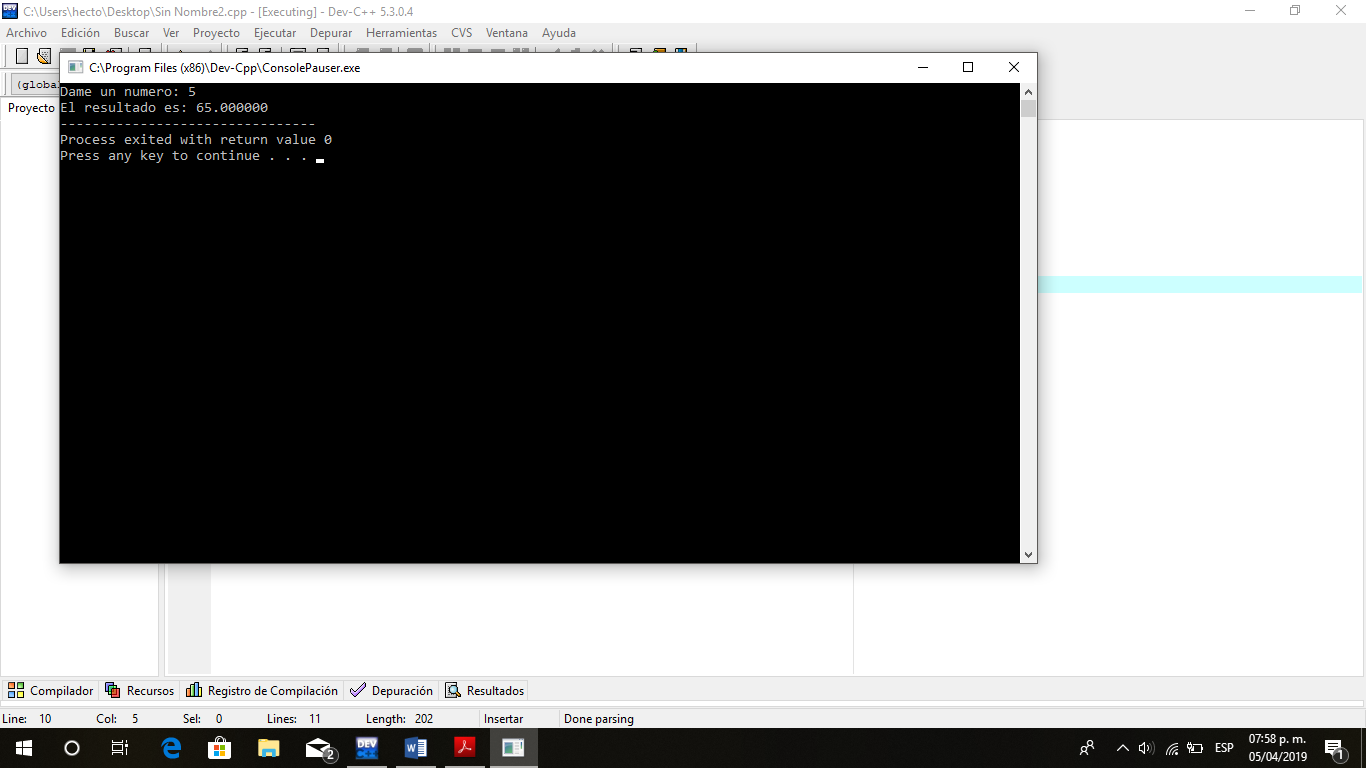


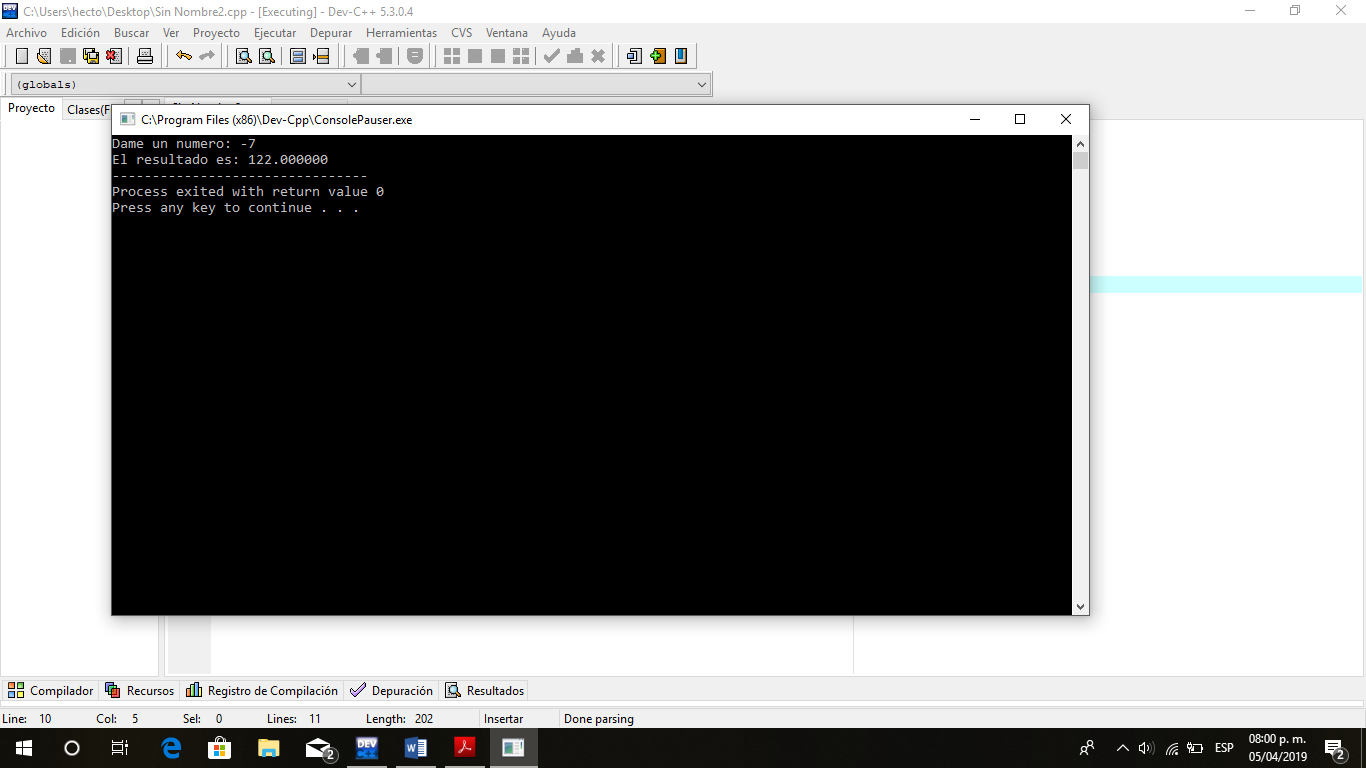
1. Menu de altas y bajas



1. Utilizando operador ternario realizer un programa que al introducer un numero mayor a dos resuelva la ecuacion 3x^+3x-25 y que cuando sea menor a dos resuelva la ecuacion 2x^2-3x+3.







**Conclusiones:**

Las estructuras de selección sirven para lograr programar cosas más complejas, en caso de necesitar que el programa ejecute uno de varios resultados posibles o que vaya por diferentes “caminos” dependiendo de los valores que tomen ciertas variables.