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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Karina García Morales |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1121 |
| *No de Práctica(s):* | Práctica no.8 |
| *Integrante(s):* | Espinosa Carrillo Luis Andrés |
|  |  |
| *No. de Equipo de cómputo empleado* | Equipo no. 20 |
| *Semestre:* | 2019-1 |
| *Fecha de entrega:* |  |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía práctica de estudio 08:**

**Estructuras de selección**

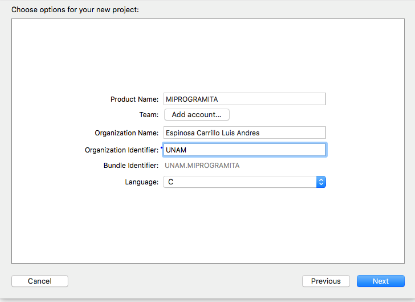
**Objetivo:**

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.

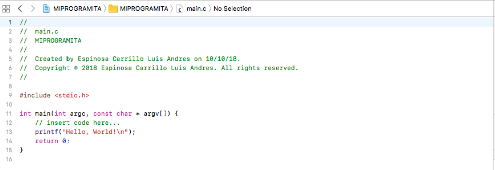
**Desarrollo:**

* 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.

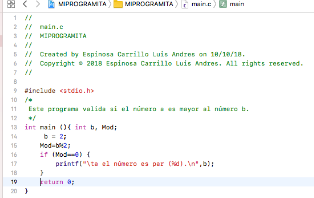
Al inicio de la práctica creamos un programa en la terminal.



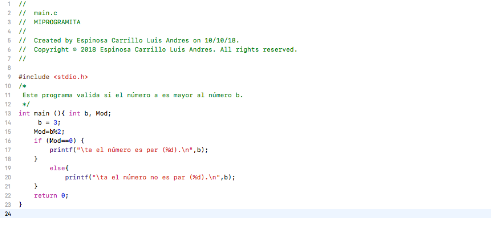
Creamos un programa que nos imprimiera el siguiente texto.



Desarrollamos un programa que nos dijera si un número que introdujéramos era un número par o un número impar.



Luego hicimos un programa que nos validara si un número a, fuera más grande que un número b o si el número b era más grande que el número a.

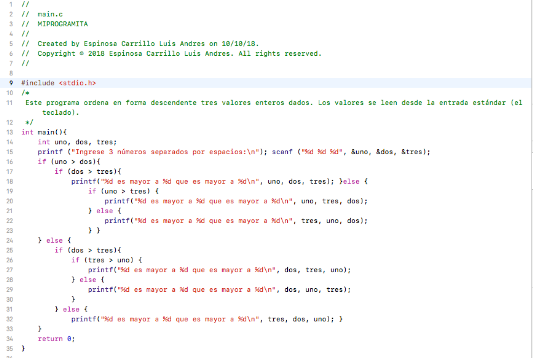


**Estructura de control selectiva if:**

La estructura de control de flujo más simple es la estructura condicional i**f**

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.

Creamos un programa que nos verificara si un número era par:



**Estructura de control selectiva if-else**:

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.

Al programa de condicional simple lo modificamos para que fuera un programa de condicional compuesta:

Le cambiamos el **if**(0) y con un número par nos decía que era impar porque no se cumplía la condición y el 0 lo toma como falso para cumplir la condición

Sin embargo; al poner **if**(-3) nos mostraba que el número era par porque se validaba la condición al tener un valor diferente de 0 se considera verdadero

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

Copiamos el programa de la página, utilizamos sangrías para anidar los ciclos **if,**y luego identificamos que era lo que hacía cada**if**y cada**else,**y veíamos donde se abrían y se cerraban cada **if**y cada **else**

Después comentamos que **printf**fue el que nos imprimió en nuestra prueba de escritorio

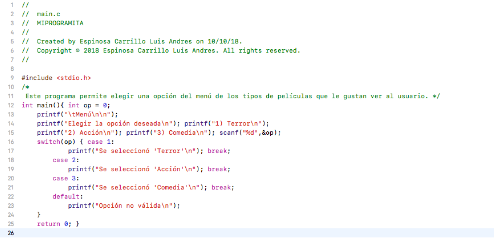
**Estructura de control selectiva switch-case**

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.

Se usan datos de tipo Entero y Carácter puesto que, al usarse principalmente como menús, nos da a elegir entre opciones enumeradas o con incisos.

En el primer programa le dimos a elegir al usuario entre 3 géneros diferentes de películas al ingresar un dato de tipo entero que coincidiera con las opciones dadas:



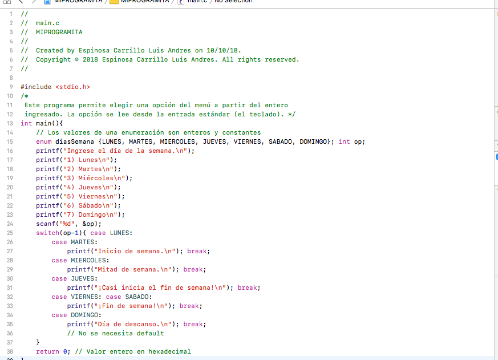
En el segundo programa se hizo la misma dinámica, simplemente le cambiamos el tipo de dato a carácter:

**Enumeración**

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).

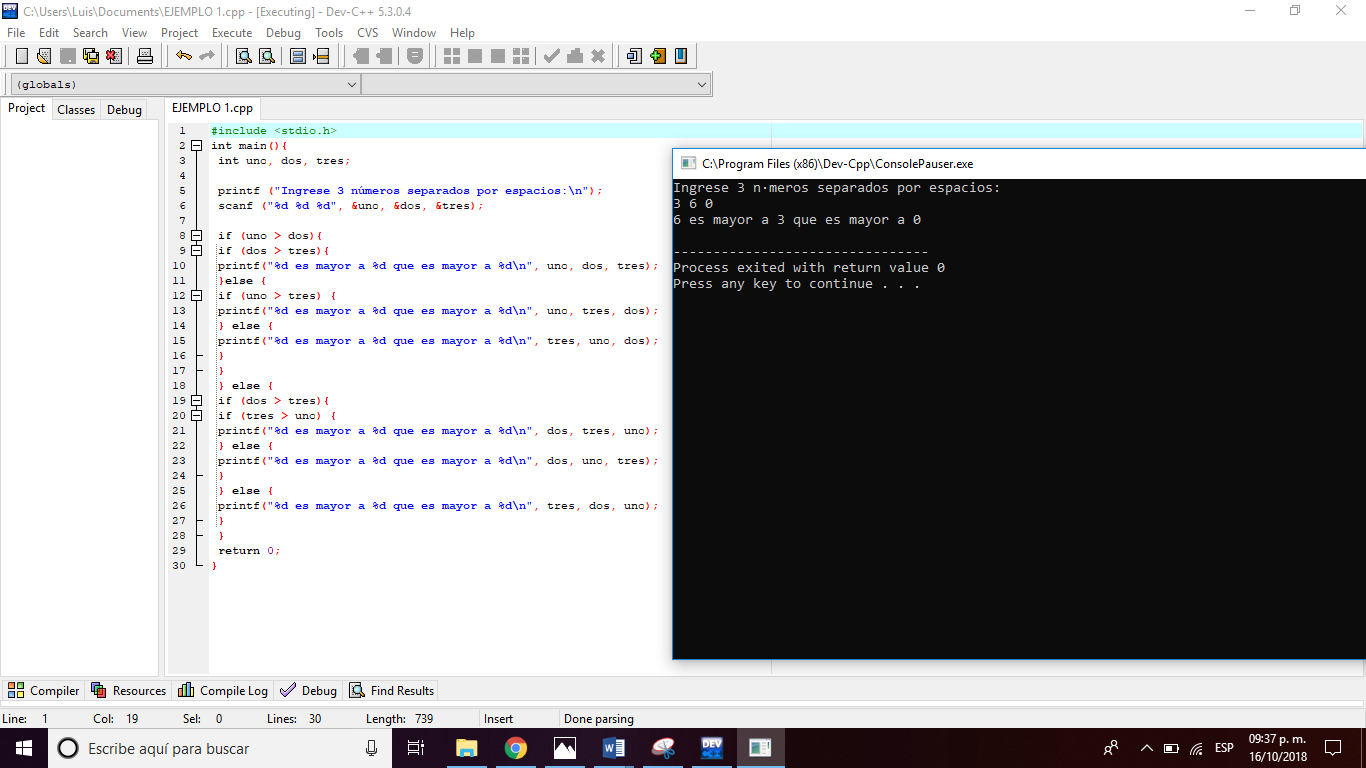
La enumeración se llama 'boolean' y contiene dos elementos, el primero (FALSE) posee el valor 0 y el siguiente (TRUE) posee el valor 1. Si hubiese más elementos en la enumeración, la numeración correría de manera ascendente.

Copiamos y ejecutamos el programa de la página 146 de la práctica



**Ejercicios de tarea**

* 1. Genera un programa que indique cuál de tres número es el mayor, emplea el condicional ternario

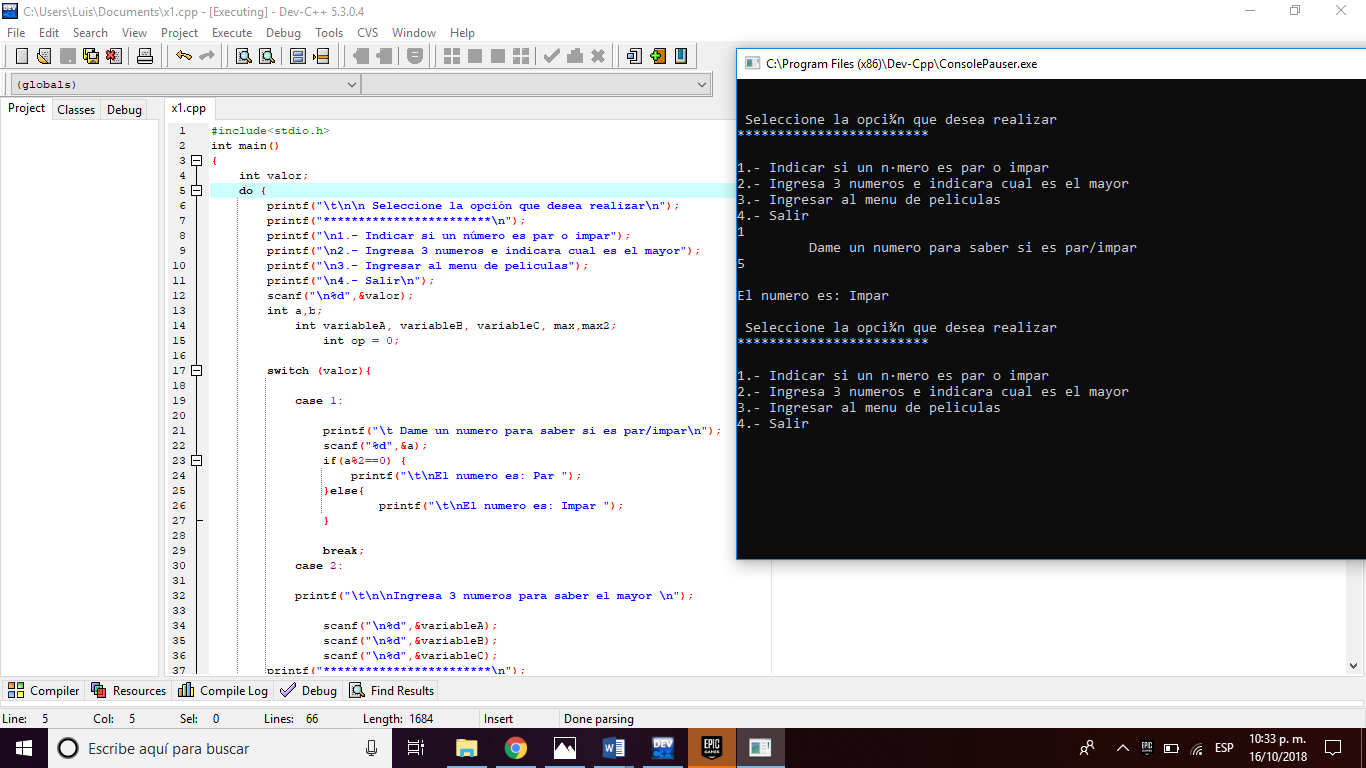


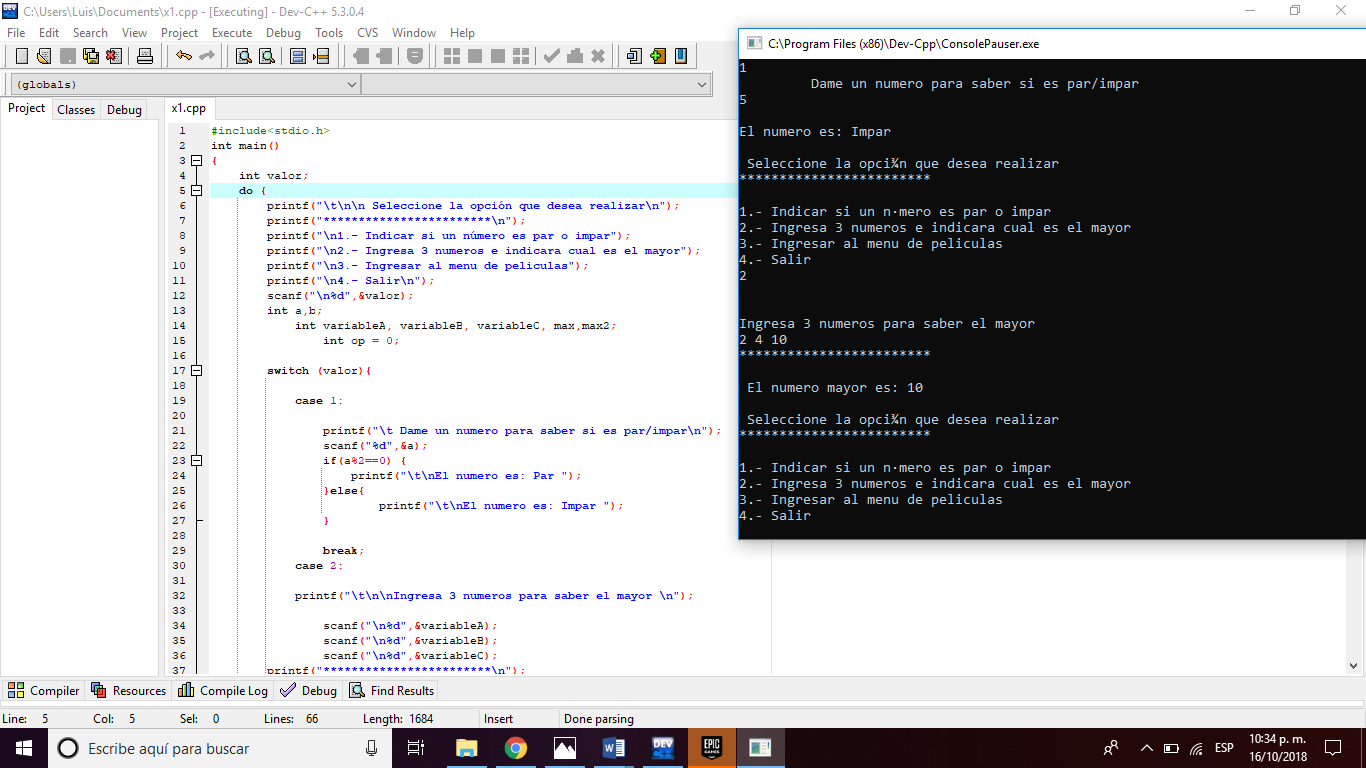
2. Genera un menú

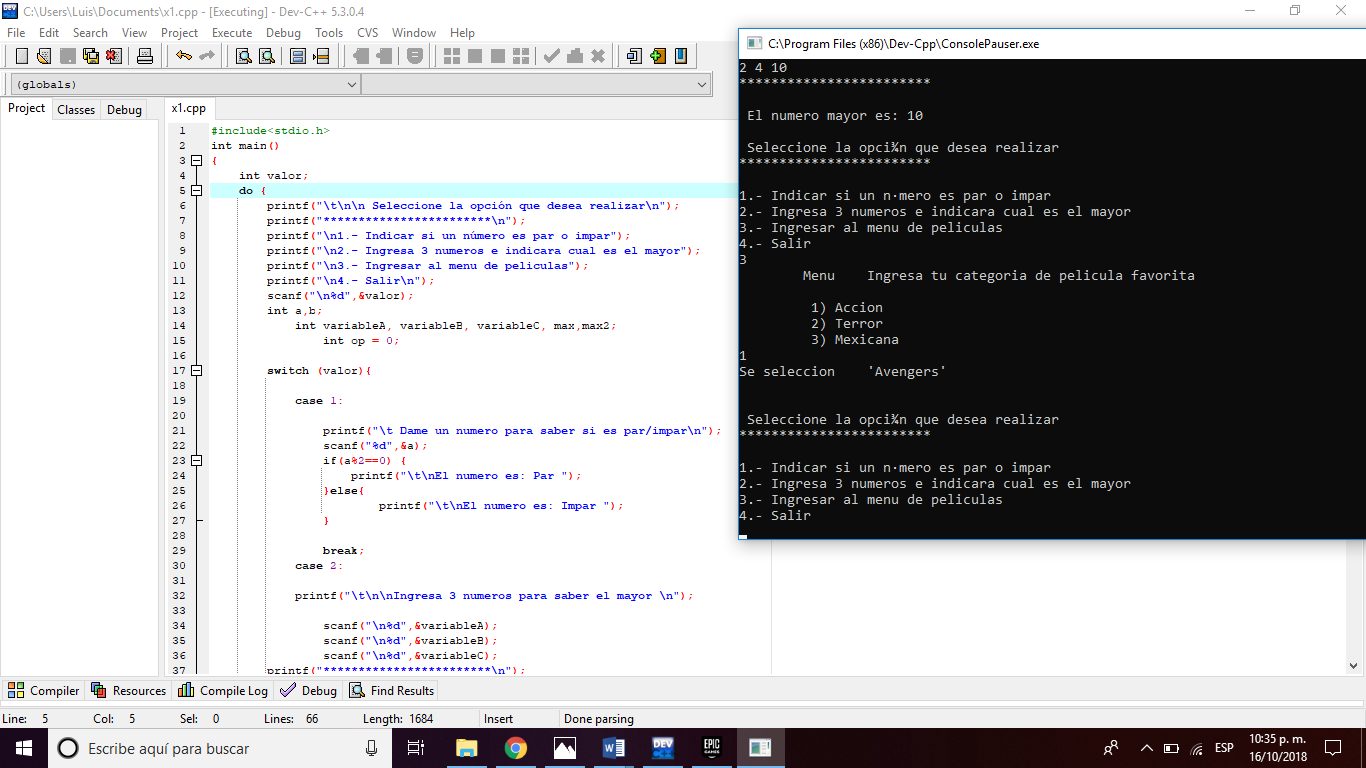
a) Ingresa un número e indicar si es par o impar

b) Ingresa tres números e indicar cuál es el mayor(programa 1)

c) Ingresa al menú de películas







**Conclusiones:**

 Estuvo muy fácil la realización de la práctica, no tuve mayor complicación al realizar el programa de los ejercicios de tarea, además que fue muy divertido hacer el desarrollo de la misma.

**Bibliografía:**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.