**INSTITUTO POLITÉCNICO NACIONAL**

**Escuela Superior de Cómputo**

**Ing. en sistemas computacionales**

**Nombre:**

**Josue Macias Castillo**

**Profesora:**

**Luz María Sánchez García**

**Grupo:**

**2CM1**

**Boleta:**

**2015301058**

**Materia:**

**Teoría Computacional**

**Fecha de entrega:**

**28 de abril del 2017**

**Introducción:**

El propósito de este documento es reportar lo acontecido en la práctica 2 en la cual se le pide al usuario que ingrese cadenas de tipo binario que concuerden con la expresión GLC dada.

**Planteamiento del problema:**

Se nos pide implementar en cualquier lenguaje de programación una solución que al ingresar una cadena esta se valida y se compare con la GLC dada previamente por el mismo programa, en este programa la cadena solo debe contener los valores 0 y 1, ya que no acepta cadenas con otros números o con letras.

**Implementación de la solución:**

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**using** System**.**Text**.**RegularExpressions**;**

**namespace** Practica\_4

**{**

class Program

**{**

//Variables globales para validar

const int valido **=** 1**;**

const int invalido **=** 0**;**

static void Main**(**string**[]** args**)**

**{**

//Declaracion de variables a usar

string cadena**;**

int n**,** m**;**

//int opcion = 0;

char**[]** caracteres**;**

string patron **=** "^([a-z]+|[2-9]+|[A-Z]+)$"**;**

Console**.**WriteLine**(**"Expresion GLC"**);**

Console**.**ForegroundColor **=** ConsoleColor**.**Yellow**;**

Console**.**WriteLine**(**" S --> 1|0|11|00|1S1|0S0"**);**

Console**.**ForegroundColor **=** ConsoleColor**.**Gray**;**

**while** **(true)**

**{**

//Da color a las letras

Console**.**ForegroundColor **=** ConsoleColor**.**Gray**;**

//El programa pide una cadena

Console**.**WriteLine**(**"Ingresa una cadena: "**);**

Console**.**ForegroundColor **=** ConsoleColor**.**DarkCyan**;**

cadena **=** Console**.**ReadLine**();**

caracteres **=** **new** char**[**cadena**.**Length**];**

//Valida si la cadena es binaria

bool validacion **=** Regex**.**IsMatch**(**cadena**,** patron**);**

//En caso de ser falso hace la operacion para comprobar si es palindromo

**if** **(**validacion **==** **false)**

**{**

caracteres **=** cadena**.**ToCharArray**(**0**,** cadena**.**Length**);**

**if** **(**verificar**(**0**,** caracteres**,** cadena**.**Length**)** **==** valido**)**

**{**

Console**.**ForegroundColor **=** ConsoleColor**.**Green**;**

Console**.**WriteLine**(**"La cadena es valida"**);**

**}**

**else**

**{**

Console**.**ForegroundColor **=** ConsoleColor**.**Red**;**

Console**.**WriteLine**(**"La cadena es invalida"**);**

**}**

**}**

//En caso contrario muestra un mensaje de error

**else**

**{**

Console**.**ForegroundColor **=** ConsoleColor**.**DarkRed**;**

Console**.**WriteLine**(**"La cadena debe ser binaria"**);**

**}**

**}**

Console**.**ReadKey**();**

**}**

//Funcion que compara las letras de inicio a fin

static int verificar**(**int pos**,** char**[]** palabra**,** int largo**)**

**{**

//Primera condición, si la palabra en su posición 0 es igual a la ultima

**if** **(**palabra**[**pos**]** **==** palabra**[**largo **-** pos **-** 1**])**

**{**

//Aquí solo nos indica que si las pos es menor que largo - pos entra e incrementa pos

//cuando llega al centro de la palabra, nos indica que es un palindrome

**if** **(**pos **<** largo **-** pos**)**

**return** **(**verificar**(++**pos**,** palabra**,** largo**));**

**else**

**return** **(**valido**);**

**}**

//Si no entra al if en donde se comaparan los caracteres

**return** **(**invalido**);**

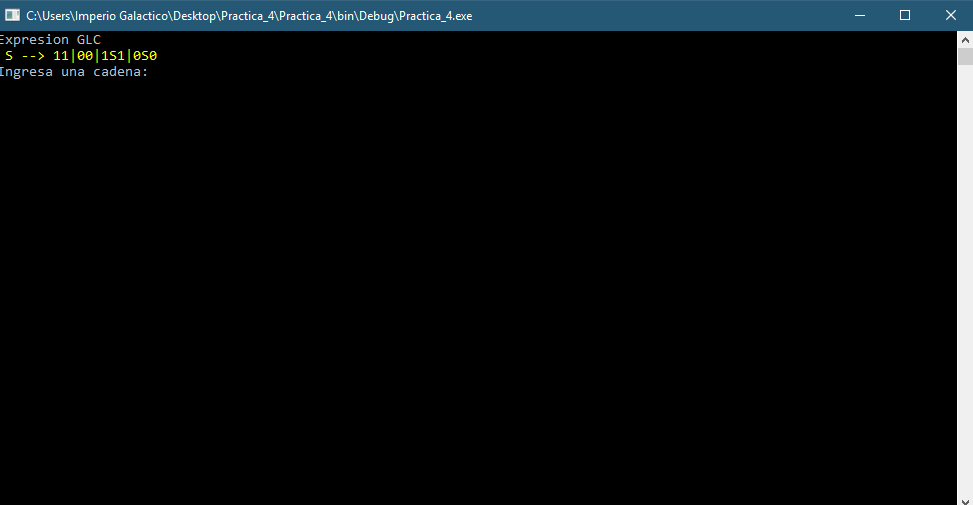
**}**

**}**

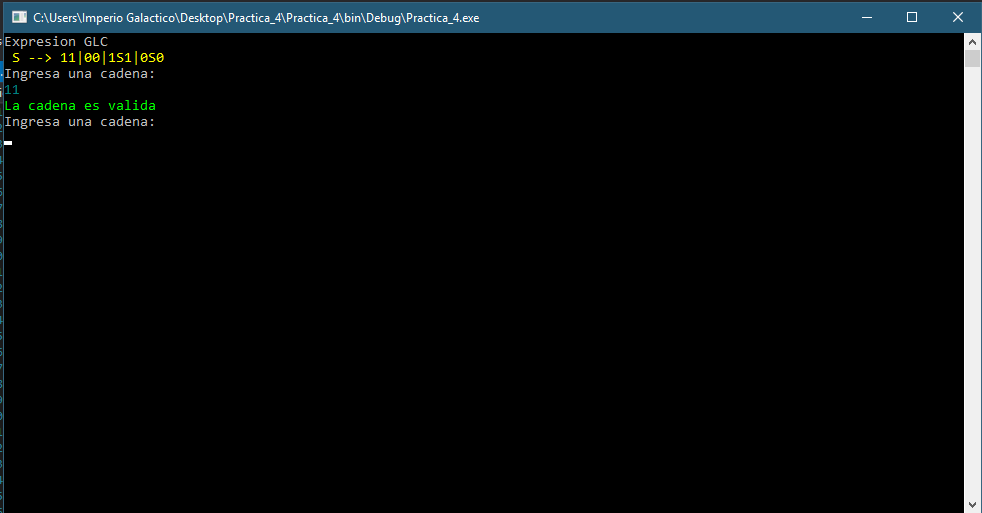
**}**

**Funcionamiento:**

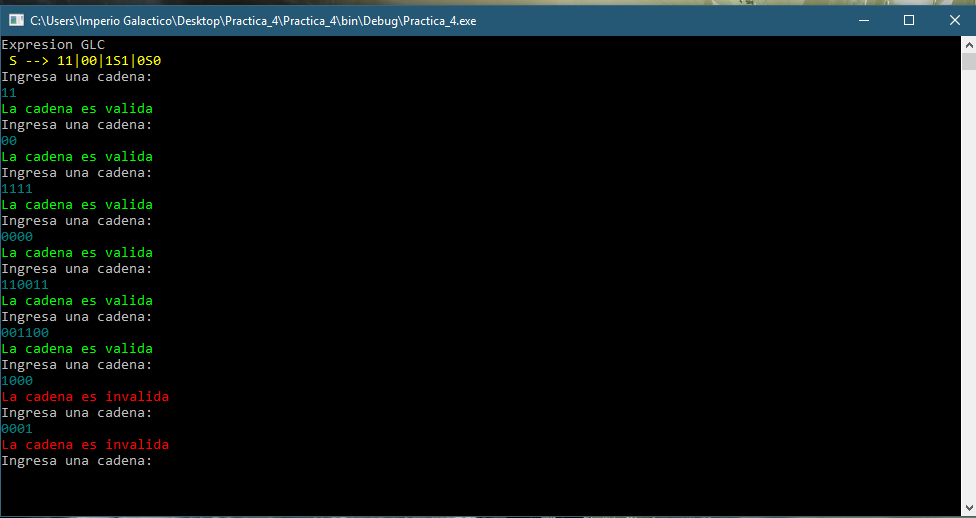
En la pantalla al principio se nos muestra la expresión GLC que está definida por el mismo programa además de que se nos pide una cadena.



Ahora le ingresamos la primera cadena que puede aceptar basándonos en la GLC



Nos muestra que la cadena es válida, ahora probemos con más cadenas.



Como se aprecia en la captura anterior la cadena probé con varias cadenas que resultaron ser válidas y al parecer se muestran 2 cadenas invalidas ya que no cumplen con la GLC.

**Conclusión:**

En esta práctica comprendí mejor el uso de las GLC además de que retome las expresiones regulares que con anterioridad había visto para poder realizar esta práctica, lo único que me falto implementar fue que el programa mostrara las transiciones en forma de árbol de cada cadena, pero con el tiempo mejorare esta práctica, aunque ya no se tenga que entregar.

**Bibliografía:**

# Bibliografía

Jeff Ferguson, B. P. (2003). *La biblia de C#.* Madrid: Anaya Multimedia (Grupo Anaya. S.A.).

John E. Hopcroft, R. M. (2007). *Teoria de autómatas, lenguajes y computación.* Madrid: PEARSON EDUCACION S.A.