Al inicializar el programa, tendremos que elegir entre 2 opciones.

La primera opción ejecutará una función realizarChequeoCadena.

//1-.

Cuando la llamemos, lo primero que va a hacer es pedir por consola una cadena para almacenarla en un array del tipo char. La función strtok va a separar esa cadena a partir de un caracter (que en este caso sería $) excluyendo este mismo y guardamos cada cadena resultante en una variable “grupoCadena”. Esto lo hacemos en un while hasta que la cadena esté vacía.

Texto

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

En cada iteración del while , se ejecutará la función distinguirConstante que recibe como parámetro una cadena y devuelve un int que nos va a indicar el tipo de automata al que pertenece la cadena .

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

Con el tipoAutomata que retorna distinguirConstante, nosotros podemos controlar con un if qué errores léxicos puede llegar a presentar.

Captura de pantalla de un celular

Descripción generada automáticamente

Los errores léxicos están a cargo de la función llamada revisarOctal/Decimal/Hexa.

Texto

Descripción generada automáticamente

Esta función recorre toda la cadena buscando caracteres que no cumplan la condición de octal, decimal Hexa. Cuando las encuentra, suma a un contador y este nos va a indicar los errores léxicos que se presentó en la cadena.

Si el grupoCadena no presentó errores léxicos, entonces se ejecutará la función verificarPalabra. Si presentó, no se podrá analizar ya que el grupo cadena no pertenecería al alfabeto.

Texto

Descripción generada automáticamente con confianza media

verificarPalabra posee un switch y éste llama a distinguirConstante para saber el tipo de autómata y con que función verificarPalabraOctal/Decimal/Hexadecimal va a trabajar.

Captura de pantalla de computadora

Descripción generada automáticamente

verificarPalabraOctal/Decimal/Hexadecimal reciben el grupo que se le pasó a verificarPalabra. Poseen un estado actual = 0 (que será nuestro estado inicial de la matriz), y verificarPalabra indica si la palabra es reconocible por el automata a través del uso de una matrizOctal/Decimal/Hexadecimal inicializada con datos de la tabla de transición del automata (está adjunto al tp) y un bucle.

Texto

Descripción generada automáticamente

Este bucle va a iterar hasta que la cadena esté vacía , guarda el carácter en una variable caracterActual y llama a la función transformarACol que devuelve 0 si caracterActual==0, 1 si caracterActual>0, 2 el resto(por ej si recibe un +, - o x para que sean usados por la matriz decimal/hexa). Con esto nos movemos a la posición 0 o 1 , (ya que 2 es imposible en esta matriz) porque la cadena no puede presentar errores léxicos debido a que verificamos todos sus caracteres antes. La matriz devuelve una letra (más que nada por comodidad) la cual nosotros convertimos a número y así va iterando todos los estados y caracteres. Si todos los caracteres estan en la posición correctas, entonces el estadoActual==2 ya que es el estado de aceptación del automata octal, con lo cual sumará al contador de cantPalabrasOctales.

Ejemplos

Texto

Descripción generada automáticamente

Cadena: 12j9

La j no pertenece al alfabeto del decimal, por lo que no se puede verificar la palabra y, por lo tanto, no hay palabras correctas Decimales.

Cadena: F0x4F

No presenta errores léxicos porque todos los caracteres pertenecen al alfabeto, pero la palabra no es reconocible por el autómata ya que x no está en la posición correcta y por lo tanto, no existen palabras correctas Hexadecimales.

Cadena: 012

No presenta errores léxicos porque todos los caracteres pertenecen al alfabeto, y la cadena es reconocida por el autómata, con lo cual la cantidad de palabras correctas octales es 1.

Texto

Descripción generada automáticamente

La primera no cumple porque ha presentado errores léxicos, la segunda tampoco porque no es reconocida por el autómata. Solo se reconoce la tercera por el autómata y por lo tanto es palabra correcta.

//3-.

Si en el menú principal elegimos la opción dos, se llamará a una función calcularExpresionMatematica();

Creamos dos estructuras pilas, PilaNumerica donde se almacenan los dígitos y los resultados (float items, del tipo float porque la división puede dar un numero con coma) de ir realizando las operaciones y PilaCaractes (char items) donde se almacenan los operadores. Además contamos con int ultPos que sirve guardar la última posición de la pila. La manera de almacenar datos en la estructura es con la función push y PushCaracteres

Texto

Descripción generada automáticamente

InicializarPila/PilaCaracteres sirve para inicializar la variable ultPos

Texto

Descripción generada automáticamente

isEmpty/EmptyCaracteres nos avisa si la pila está vacía comparando ultPos con -1, ya que cada vez que un elemento se elimina, se resta uno a esa variable, resultando siempre que esté vacia, con ultPos en -1.

Texto

Descripción generada automáticamente

pop/popCaracteres primero verifica si la pila no está vacía (con isEmpty), si lo está retorna -1 pero si no, retornará el elemento al tope de la lista y restará 1 a la última posición.

Texto

Descripción generada automáticamente

Push almacena elementos en la pila correspondiente y suma 1 a la ultima posición ´

Texto

Descripción generada automáticamente

Lo que hace la parte de operar los caracteres numéricos de la cadena es la función realizarOperacion que recibe 3 parámetros, 2 números previamente transformados con la función valorNumerico y un operador que será tomado como parámetro por un switch ya que esto nos dará la oportunidad de realizar suma, resta, división y multiplicación de los 2 números y retornar su resultado.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ahora tenemos la funcion evaluarExpresion que recibe la cadena que pasamos por consola. Creamos las pilas donde guardamos nuestros números y caracteres y los inicializamos.

Texto

Descripción generada automáticamente

Además, contamos con dos bucles:

El primero va a iterar con la condición de que la expresión[i] no sea nula (esto lo hacemos para que recorra todo el array de caracteres). Guardamos ese dato en una variable número convertido en un numero con la función valorNumerico y en una estructura de control verificamos si está dentro de 0 y 9 (esto lo hacemos para verificar que sea un digito positivo) ya que si cumple con esa condición se almacenará en la pilaNumerica.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

Pero si no la cumple eso quiere decir que es operador, por lo cual tenemos que saber qué operador es para poder controlar la precedencia.

Verificamos la de menor precedencia primero



Chequeamos que la cadena no esté vacía (con un controlador) y que la última posición de la pila sea un +, -, \* y /. Trae los dos números desde el tope de la pilaNumerica y el operador que está actualmente en el tope de pilaCaracteres. Llama a la función realizarOperación para que calcule el resultado. Por último almacenamos ese resultado en la pilaNumerica para continuar con el resto de operaciones.

En cambio, si la pila está vacía, entonces guardará ese operador en el tope de la pilaCaracteres.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Por otro lado chequeamos si llega un \* o /, que son los caracteres con mayor precedencia



entonces se chequea que no haya un \* o / pendiente (ult posicion ) ya que si los hay primero se realizará esa operación que hay en la pila (por la precedencia de izquierda a derecha). A continuación se realiza el mismo procedimiento que en caso de ser + o – .

Texto

Descripción generada automáticamente

Si en cambio la pila esta vacia o la ultima posicion de operadores contenía un más o menos se guarda directamente en la pila de caracteres el \* o / que llego.

El segundo bucle dentro de evaluarExpresion solo se encarga se realizar las operaciones que fueron desplazadas por ser de menor precedencia y cualquiera que haya quedado pendiente.

Texto

Descripción generada automáticamente