a) Descripción y análisis del problema

La necesidad principal en esta situación es el querer conocer el resultado de una operación matemática que pueda contener múltiples operaciones matemáticas de suma y multiplicación. Los problemas con los que nos encontramos son:

- Necesitamos que el programa solicite la introducción de un texto que contenga una operación matemática que puede tener sumas, restas, y puede estar ordenada con paréntesis.

- El programa no debe aceptar caracteres diferentes a los permitidos (letras, comas, etc.).

- El programa no debe aceptar combinaciones no permitidas de signos y símbolos (++,(+, etc.).

- El programa no debe aceptar un paréntesis abierto que no fue cerrado, o un paréntesis de más. - El programa debe ignorar o eliminar los espacios, retornos de línea o tabuladores.

- El programa debe mencionar en un texto cual fue el error (en caso de existir).

-El programa debe agrupar los elementos de la lista de texto, para poder definir los números, números decimales, y símbolos.

- El programa debe convertir cada uno de los datos que tienen un formato de texto, a un formato numérico, para que de esta manera permitan su operación matemática.

- Una vez transformada la información a un formato matemático, el programa debe poder resolver la operación.

- El programa debe resolver la operación, respetando los puntos decimales y respetando la jerarquía de operaciones.

-El programa debe regresar o mencionar, cual es el resultado de la operación matemática.

b) Descripción de la solución propuesta

La solución a los problemas ya mencionados consiste en una función o programa que contenga múltiples funciones que den solución a cada uno de los problemas específicos. 1. El programa comienza solicitando que se introduzca la expresión matemática, de la cual se desea conocer su resultado.

2. El programa valida que la expresión introducida es correcta y permitida.

a. Primero se verifica si la lista es vacía con la función isempty, si es así, de devuelve un mensaje de error, explicando cual fue el error.

b. Se eliminan los espacios, tabuladores y retornos de línea con la función string.

c. Para solucionar el problema de los paréntesis, se contabilizan el número de paréntesis abiertos y cerrados. Si hay más paréntesis cerrados que abiertos, es invalido. Los abiertos suman 1 y los cerrados restan 1, por lo que la suma debe dar 0, de lo contrario también es invalido.

d. De igual manera se verifica si se ha dado un texto el cual no contiene números con la función if all.

e. Se verifica si existen caracteres inválidos, comparando con los caracteres que se enlistan como permitidos. De igual manera se verifica si hay operadores consecutivos prohibidos.

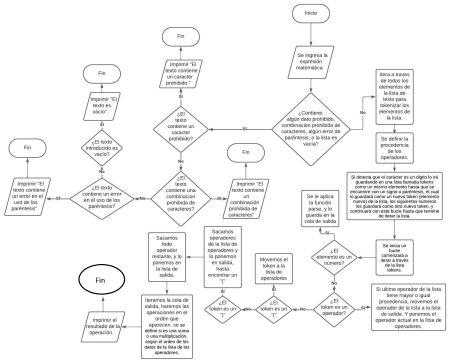

Figure 1: Enter Caption

f. Si el texto pasa positivamente las verificaciones, es válida.

3. Una vez validada la expresión, el programa necesita agrupar los elementos ya que, al estar en un formato de texto, el programa aun no distingue que conjunto de caracteres deben considerarse como un solo elemento. Por tanto, el programa comenzará a iterar por toda la lista y si detecta que el carácter es un digito lo irá guardando en una lista llamada tokens como un mismo elemento hasta que se encuentre con un signo o paréntesis, el cual lo guardará como un nuevo token (elemento nuevo) de la lista, los siguientes números los guardará como otro nuevo token, y continuará con este bucle hasta que termine de iterar la lista.

4. Ahora que ya están tokenizados los elementos en la lista tokens, un bucle comenzará a iterar a través de la lista tokens. Si el programa detecta que un elemento es un número lo le aplicará la función parse (para convertirlo en un tipo de dató algebraico) y además lo agrega a la cola de salida; de lo contrario lo guardará en la lista llamada operator-stack. 5. Posteriormente si el token es un operador, saca operadores de la pila y lo mueve a la cola de salida. Si el token es un paréntesis abierto, lo agrega a la cola de salida. Si el token es un paréntesis cerrado, saca operadores de la pila a la cola de salida. Posteriormente se saca el paréntesis de apertura de la pila. Por último, saca cualquier operador restante de la pila a la cola de salida. 6. Finalmente se utiliza una función para evaluar la función, la cual itera a través de la cola de salida y va operando binariamente los números, se define si es una suma o una multiplicación, según el orden de los datos de la lista llamada operator-stack.

c) Algoritmo

d) Diagrama de flujo

e) Instrucciones para utilizar el programa

Descripción: Programa que calcula el resultado de la operación matemática que sea introducida. Instrucciones: Al llamar a la función o programa, se te solicita introducir la expresión matemática que de la cual se desea conocer el resultado, la expresión puede estar conformada por sumas y/o multiplicaciones

de números, y que además que pueden contener paréntesis. Los únicos caracteres permitidos son:

- Dígitos

-Los símbolos: + ∗ . ( )

- Espacio, tabulador, retorno de línea

Se ocasionará un error, si:

- La expresión introducida es vacía o si está conformada únicamente por espacios, tabuladores, o retornos de línea.

- Se utilizan combinaciones no permitidas de signos como: "++", ")+", "+*"

- Algún paréntesis fue abierto y no fue cerrado.

- Algún paréntesis fue cerrado y nunca fue abierto (hay algún paréntesis de más).

- No hay ningún dato dentro en un paréntesis ()

f) Nombres de los autores

- Mateo Quirós Asprón
- Diego Martínez Mendoza
- Rodrigo Campero De la Peña
- Juan Alberto Mejía Consospó

g) Porcentaje de contribución de cada autor, según la estimación de la mayoría del equipo.

- Mateo Quirós Asprón (33.3%)

-Diego Martínez Mendoza (0.0%)

-Rodrigo Campero De la Peña (33.3%)

-Juan Alberto Mejía Consospó (33.3%)