



Instituto Tecnológico de Costa Rica

Profesor: Eduardo Adolfo Canessa Montero

Tare Programada 1

Estudiantes
Ian Murillo Campos
Julián Rodríguez Sarmiento

Entrega: 15 de mayo, I Semestre 2023

Introducción al trabajo:

Se busca realizar un código en Lenguaje ensamblador NASM X86_64 el cual sea capaz de resolver una ecuación enviada por un usuario, la ecuación puede contener variables a las cuales el mismo usuario puede asignarles un valor y se retornará en consola el resultado de la ecuación.

La ecuación puede ser por ejemplo $2X + (4Y * 3) / Z$ y seguidamente los valores de X, Y y Z los dará el usuario antes de comenzar a resolverse el algoritmo, por lo que la entrada del programa podría ser:

$2X + (4Y * 3) / Z,$

$X = 4,$

$Y = 5,$

$Z = 8$

y se desplegará como resultado 76

Procedimiento para la resolución:

Para realizar el código solicitado se optó por utilizar una implementación de un algoritmo que resuelva expresiones postfijas, reordenando la ecuación de entrada a una ecuación postfija y realizando sus debidos pasos para resolverla, de esta manera se logra obtener el resultado más fácilmente, sin pasos extra ocasionados por prioridades como por ejemplo algún paréntesis.

Esto se logra reordenando la expresión manteniendo al final los operadores de la ecuación y al inicio los operandos, por ejemplo $2+4$ pasaría a ser $2\ 4\ +$ siguiendo una serie de reglas y utilizando operaciones con pila para ordenar la ecuación de forma que se evalúen los elementos con prioridad (por ejemplo paréntesis) de la misma manera que se realizan las demás operaciones, cuidando siempre el orden de prioridades, para resolver la ecuación una vez ordenada de manera postfija, al encontrarse un operador se realiza su función a los 2 operadores anteriores a este, sustituyendo dicho resultado en la ecuación original para repetir el proceso hasta que se encuentre un único operando el cual es el resultado.

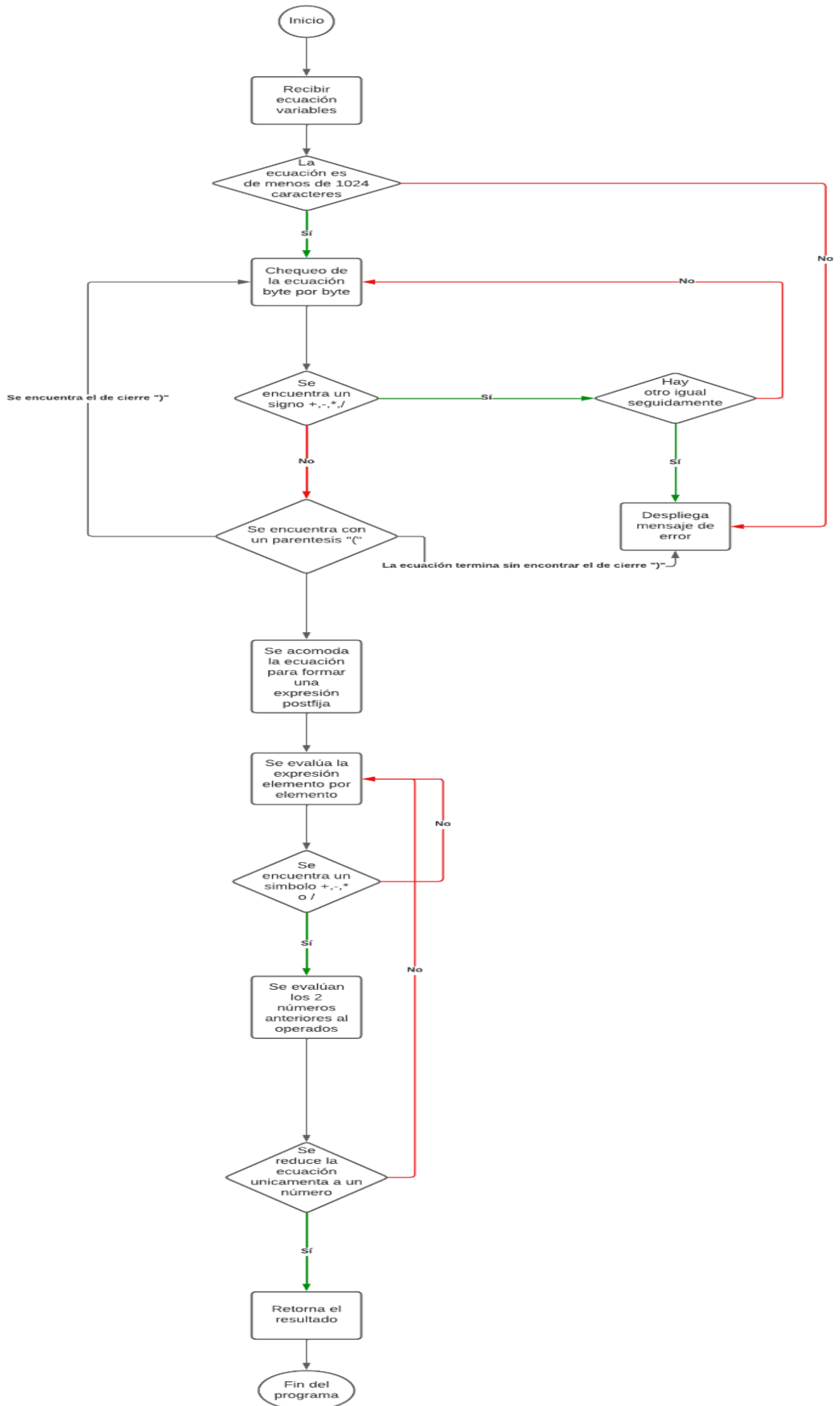
Por ejemplo, la ecuación $2+(4*3-2)$:

Pasaría a ser postfija a la siguiente ecuación: $2\ 4\ 3\ * \ 2\ - \ +$

Al encontrar el primer operador (*), se le realizará la multiplicación a los dos operandos encontrados antes (4 y 3) dando como resultado ya con sustitución la expresión: $2\ 12\ 2\ - \ +$

Siguiendo con los pasos se encontrará el (-) el cual se aplica para el par (12 y 2), dando como resultado la expresión: $2\ 10\ +$, para finalizar aplicando el (+) al par (2 y 10) dando como resultado el 12, siendo la solución de la ecuación.

Diagrama de flujo:



Manual de usuario:

El programa solicitará al usuario una ecuación seguida de los parámetros de la misma, estos datos deberán darse en el siguiente formato: (ecuación “,” parametro 1 “=” valor del parametro, parametro 2 “=” valor del parámetro 2) y así con hasta un máximo de 10 parámetros (tomar en cuenta que los parámetros deben estar con letra minúscula e importante colocar una coma “,” al final de la ecuación y de cada variable a no ser que ya no se quieran añadir más variables), se puede visualizar mejor en el siguiente ejemplo:

$2x + 4y(8/2z),$

$x = 7,$

$y = 5,$

$z = 1$

Al ingresar los datos, el programa procederá con la revisión de los mismos, si existe algún error con el formato, se desplegará un mensaje de error, si este no es el caso, comenzará a realizar la sustitución de las variables y evaluar la ecuación para así proceder a mostrar el resultado en pantalla.

Conclusiones:

Lastimosamente no se logró concretar el funcionamiento para evaluar las operaciones en el código.

Las Ecuaciones se revisan perfectamente, encuentra errores en la cantidad de paréntesis, el número de variables y formas válidas de digitar los signos en la ecuación. Todo esto antes de procesar la ecuación.

Hubo problemas a la hora de reescribir la ecuación postfija después de evaluar los primeros operandos.