**Inputs**

* 2 string arrays from keyboard

**Outputs**

* Message asking for user to input 2 string arrays
* Message displaying final answer Levens Distance
* Message displaying final answer Hammings Distance
* Message asking if user wants to continue program or exit

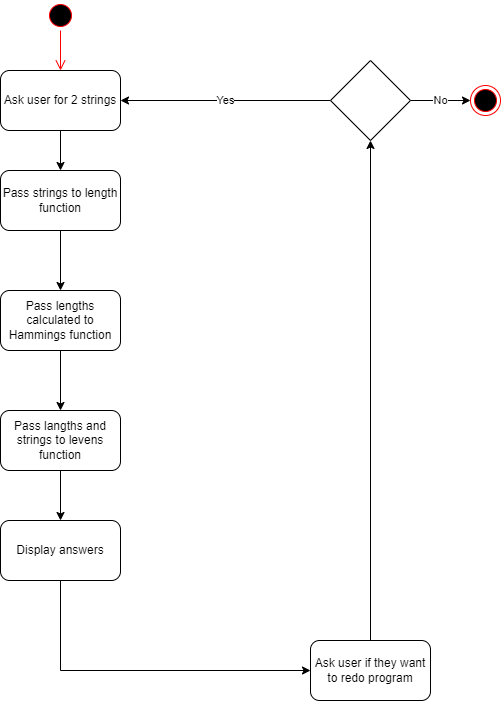
**Variables**

* Output – BYTE for saying display function final hamming answer: msgDistHamming
* Output – BYTE for saying display function final leven answer: msgLeven
* Output – BYTE for if string array lengths differ: msgLength
* Output – BYTE message for input string array 1: msgInputStr1
* Output – BYTE message for input string array 2 : msgInputStr2
* Output – BYTE for asking user if they want to exit program: msgExit
* Output – BYTE for formatting, a line: msgLine
* Input – BYTE Array of characters from keyboard: str1
* Input – BYTE Array of characters from keyboard: str2

**Algorithm**

1. Ask user to input 2 strings
2. Call a function to get the length of the 2 strings
3. Call a function to implement hammings distance (how dissimilar the 2strings are)
4. Display answer
5. Call a function to implement levens distance
6. Display answer
7. Ask user if they want to exit or repeat program

**Flow Diagram**



**Stack Diagrams**

**Get Length Function**

|  |
| --- |
| String pointer |
| return address |
| Old EBP |
| EBX |
| ECX |
| EDX |
| FLAGS <-ESP |
|  |

**Leven function**

|  |
| --- |
|  |
| String length 1 |
| String pointer 1 |
| String 2 length |
| String pointer 2 |
| return address |
| Old EBP |
| EBX |
| ECX |
| EDX |
| FLAGS <-ESP |
|  |

**Hammings function**

|  |
| --- |
|  |
| 0 |
| 1 |
| String length 1 |
| String length 2 |
| return address |
| Old EBP |
| EBX |
| ECX |
| EDX |
| FLAGS <-ESP |