**Interpreter**

**Section 1: "General Description"**

This project allows you to do arithematic operations with no difficulty till you stop it and then displays to you the variables you entered with their calculated values …. It also points out all the errors you may not consider to let you work on it correct and safe …..

**Section 2: "Program Structure"**

**-First "Main Function":-**

This function is clear and easy to run this simple program . It is ordered in it to display for the user a statement to enter his expression and to write the word exit after finishing then after calling all the coming functions with all their uses and benefits ….

The function printlinkedlist is used to display for the user all what he wished to see on his screen from variables and calculations correctly ….

And a statement that sounds an error will be displayed if there is something wrong entered by the user that is not logically valid ..

This was put in consideration during handling the errors ….

\*-A structure containing name and data points to an array of structs at which all data will be stored in .

\*- Implementation of stacks using linked lists …

**-Second "addtail Function":-**

Adds a new node at the end of the array

**-Third "searchlinkedlist Function”:-**

This function searches for a variable entered by the user if it is found before … this function can use this repeated variable to do any operation on it as the user wishes correctly with no limitations ….

**-Fourth "isdigit Function":-**

Checks if the entered character by the user is a digit or not …

**-Fifth "ischar Function":-**

Checks if the entered character is a letter from a to z or not …..

**-Sixth "isoperator Function":-**

Checks if the entered character is an operation or not …

**-Seventh "piriority Function":-**

Checks the piriority of the operations and the brackets as well to do the calculations

correctly as wished to be …

**-Eighth "infixtopostfix Function":-**

This function changes the expression entered by the user into postfix expression to be easily evaluated … Pariority function is called to check the right pariority of operations used considering the floating point and the negative for the number at any place inside the entered expression and the 2 digit numbers .. The postfixed expression is stored in a string

**-Nineth "evaluate Function":-**

This function takes the operands and the operation and do the right operation on the 2 operands .. This will be useful in the evaluation of the postfixed expression !

**-Tenth "evaluatepostfix Function":-**

This function evaluates the postfixed expression after checking in all the errors and can calculate floating numbers and negative numbers …. And return its value …

**-Eleventh "newvariable Function":-**

This function calls almost some of the previous functions

It stores the postfixed expression in a string and evaluates the expression and store it in a float variable and searches for the repeated variables if found , if there is no repeater variable it uses the function addtail to add new node at the end of the array and if not it

continues adding data to it ….

**-Twelveth "printlinkedlist Function":-**

This function to display all what the user has made and wants to calculate at the end of his trial to the program …

**Features**

**Handling the errors :**

This is well done in the function (checkvalid)

It runs as long as the expression entered has letters and numbers and operations using the previous functions ….

It points out an error for the user when :

\* the user enter 2 equals (==)

\*enters an undefined character ($ , !)

\*entering a variable name that is not defined before .

\*entering a negative in a wrong place

And any other msitakes can be done by the user not mentioned is considered as well ….

**User guide and experience :**

This program is really entertaining and helpful for you to do operations as you wish with non stop and find the results clear and right …

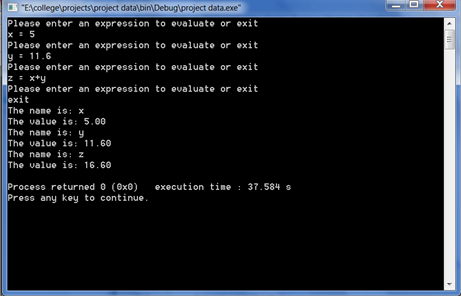
\*First : you will be asked to enter your expression and it will not end up it keeps on calculating all what you wants till you write exit ..

\*Second : Be careful not to confused with wrong shapes on your keyboard or the program will stops finding out an error ..

\*Third : write (exit) if you want to display the results of your work clealrly

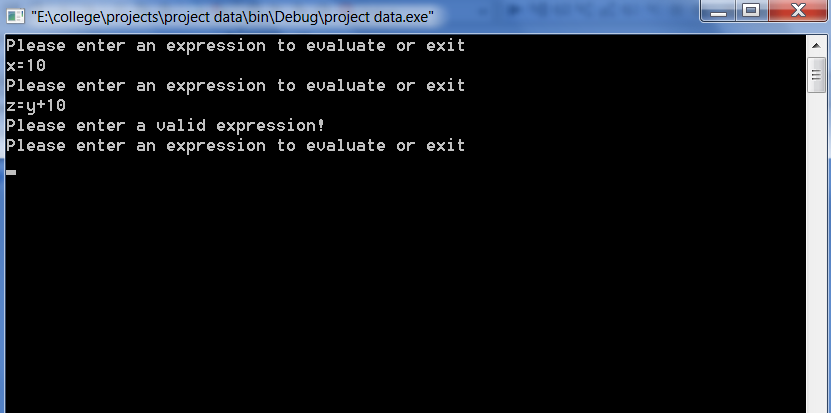
It is easy and clear and can help you a lot with your work ☺ …..

**Screenshots for running the program**



**\*\* Some errors**

**Entering a variable not used before**



**Entering a double equal OR entering a strange character**

