Reasons for not able to finish Assignment-2:

Regarding the question asked I know what algorithm to follow to solve the question but the only reason I was not able to finish the assignment is my lack of familiarity with C programming language. I am very comfortable programming in JAVA and Python. However I was unable to do the same in C as there are differences in not the syntax of the commands that mattered but different in-built functions which according to my opinion Python and Java are better in that particular area in comparison to C. For example in Python, an array can be disintegrated into number of arrays using delimiters however I was not able to find that type of function for C(may be that type of function is present in C but I was unable to find it). I know programming is essential but I request you to please consider the Algorithm also.

ALGORITHM:

- ➔ First of all read the expression as a string or array.
- ➔ Now disintegrate the string or array into multiple arrays using the operators as delimiters.
- ➔ Hence we have an array consisting of elements which are operand and operator consecutively in that specific sequence.
- ➔ Now we create two stacks named value and operator.
- ➔ We initiate a token and make the token read the elements of the array in order.
- ➔ If the token reads a number then it pushes that particular number into the Stack named Value.
- ➔ If the token reads an operator then the token checks if the Stack operator is empty or not. If the Stack operator is empty then it pushes that particular operator into the Stack operator. And if the Stack operator is not empty then it compares the precedence of the operator present at the top of the Stack operator with the currently read operator.
- ➔ If the precedence of the operator present at the top of the Stack operator is higher or equal to the currently read operator then we pop two numbers from the Stack value and the top operator from the stack operator and perform the operation with that particular operands and operator. Note that the order of the operand is maintained while performing the operation. Thereafter we push the result of the operation into the Stack value. However if the precedence of the operator present on the top of the Stack operator is less than the currently read operator then we simply push the currently read operator into the Stack operator.
- ➔ The token follows the above algorithm for all the elements that it reads and at last the Stack operator is empty and the Stack value possesses only one element which is the answer to the expression entered as the question.

  I will try my best to complete the assignment next time.
  Sorry.