Task1.1

a. Infix expression to prefix.

Step 1:

Reverse the infix expression .

Step 2:

Scan Expression from right to left .

Step 3:

If an operand is encountered add it to output

Step 4:

If a right parenthesis is encountered push it onto STACK

Step 5: If an operator is encountered then:

a. Repeatedly pop from STACK and add to output each operator (on the top of STACK) which has same or higher precedence than the operator.

b. Add operator to STACK

Step 6: If left parenthesis is encountered then

a. Repeatedly pop from the STACK and add to output (each operator on top of stack until a left parenthesis is encountered)

b. Remove the left parenthesis.

b. Infix expression to postfix.

Step 1 :

Start reading the infix expression from left to right.

Step 2:

If we scan a operand we output it, print it. Else, If the scanned operator is greater in precedence than the operator in the stack or if the stack is empty or the stack

contains a "(", push it.

Step 3:

Else, Pop all the operators having greater or equal precedence

that of the scanned operator. After doing that Push the scanned operator to the stack. In case there parenthesis while popping then stop and push the scan operator in the stack.

Step 4:

If a '(' is encountered, push it onto Stack.

If a ')' is encountered, repeatedly pop from Stack and output it

until a '(' is encountered.

Step5: Repeat Step 2 to 4 for each element until the Stack is empty.

.