***NAME → SHIVAM KUMAR***

***ID → 201551087***

***LAB 5 (Introduction To Algorithm)***

***COMMENTS →***

***Class*** *→ (PostfixPrefixUsingArrayTree) →This is main class in which I am taking input as infix from user and calculating postfix and prefix.*

***Switch*** *cases 1 →this case is used to calculate prefix of the given infix .*

*2 → this case is used to calculate postfix of given infix.*

*3 →this case is used to calculate postfix of given prefix.*

***Class*** *→( PostfixPrefixUsingTree) → This class actualy calculates prefix and postfix via having access to TreeByArray class.*

***Methods*** *→( inToPre()) → with help of this method we get prefix from infix. In this I have used switch to check for parenthesis and operator via gotOperator() and gotParenthesis() respectively. Using prefix (root,left,right) criteria with help of tree i.e, By adding higher priority operator in right side of previous operator to right of tree of root operator and else, if it is of less priority ,then root equals next operator of lower preority and left of root equals the actual previous root and so on as in binary search tree.*

*inToPost()// →with help of this method we get postfix from infix. In this I have used switch to check for parenthesis and operator via gotOperator() and gotParenthesis() respectively. Using prefix (left,right,root) criteria with help of tree i.e,By adding higher priority operator in right side of previous operator to right of tree of root operator and else, if it is of less priority ,then root equals next operator of lower preority and left of root equals the actual previous root and so on as in binary search tree.*

*preToPost()// → with the help of this method we calculate postfix of given prefix .It is optional.*

*gotOperator(char opThis,int prec1,char x)// →with the help of this method tree is made of operator by adding higher priority operator in right side of previous operator to right of tree of root operator and else, if it is of less priority ,then root equals next operator of lower preority and left of root equals the actual previous root and so on as in binary search tree.*

*gotParenthesis(char x)//→ this method is used when a '(' comes then a while loop is used to check the end ')'parenthesis.(Skipping the "("and")"operators that comes in the mid of start"("and end ")").*

***Class*** *→( TreeByArray)→ In this class I am taking operators and making it in tree type feel. By adding higher priority operator in right side of previous operator to right of tree of root operator and else, if it is of less priority ,then root equals next operator of lower preority and left of root equals the actual previous root and so on as in binary search tree.*