// Function definition of infixToPostFix with one parameter being passed by value

// Local object with template type char to connect object to the members of the header files

// Local object stk uses dot notation to call member function push

// Local int declared to hold the value of the length of the infix being passed through the function by value

// Local string declared to hold the post fix notation after being converted

// Local char variable to hold characters read from the file

// For loop to read through the value stored in the string sLength

// if statement checking for validity of the characters being stored in the infix variable

// if the characters from infix are valid then they are stored one at a time in to the pstFix variable

// places spaces between each character for proper output

// else if statement checking for open parentheses

// if character is an open parentheses it uses the object stk and the member function push to push the ( onto the stack

// else if statement checking for close parentheses

// while loop that continues until the top of the stack is either a N or an (

// char variable hold the top char of the stack

// pops the character off the stack

// hold is stored into pstFix string variable

// adds whitespace to pstFix for proper output

// if statement for when an open parentheses is encountered at the top of the stack

// open parentheses is stored into the hold variable

// pops the open parentheses off the stack

// else statement

//while loop with function call to prec to check for precedence of character on the stack

// hold stores the character using the stk object and the member function top

// uses the object stk and member function pop to pop character off stack

// hold is stored into pstFix string variable

// adds whitespace to pstFix for proper output

// object stk uses member push to push each individual character stored in infix onto the stack

// while loop that continues until the flag value at the end of the stack is reached

{

hold = stk.top(); // hold stores the character using the stk object and the member function top

stk.pop(); // uses the object stk and member function pop to pop character off stack

pstFix += hold; // hold is stored into pstFix string variable

pstFix += " "; // adds whitespace to pstFix for proper output

}

cout << pstFix << endl; // outputs the post fix notation to the console

fout << pstFix << endl; // outputs the post fix notation to the output file