06/09/25 Lab-3

```
#include <stdio.h>
# include <ctype.h>
# define MAX 50

char infix [MAX];
int top = -1;
char stack [MAX];

void push (char c);
char pop ();
char peek ();
int precedence (char element);
int associativity (char element);
void infixtopostfix (char infix [], char postfix []);

void push (char c) {
    if (top == MAX-1) {
        printf ("Overflow \n");
    }
    else {
        stack [++top] = c;
}
char pop () {
    if (top == -1) {
        printf ("Underflow \n");
        return -1;
    }
    else {
        return stack [top--];
    }
}
```

```
char peek () {
    if (top == -1) {
        return -1;
    }
    else {
        return stack [top];
    }
}
int precedence (char element) {
    if (element == "*" || element == "/" ||
        element == "+" || element == "-") {
        return -1 ;
    }
    if (element == "^") {
        return 2 ;
    }
    return 0;
}
int associativity (char element) {
    if (element == "^") {
        return 1;
    }
    return 0;
}
void infixtopostfix (char infix [], char postfix []) {
    int i, p=0 ;
    char c ;
    for (i=0 ; infix [i] != "\0" ; i++) {
        c = infix [i] ;
    if (isalnum (c)) {
        postfix [++] = c ;
    }
    else if (c == "(") {
        push (c) ;
    }
    else if (c == ")") {
        while (peek () != "(") {
            postfix [p++] = pop () ;
        }
        pop () ;
    }
    else {
        while (top != -1 && (precedence (peek ()) >
            precedence (c) || (precedence (peek ()) ==
            precedence (c) && associativity (c == 0)))){
            postfix [p++] = pop () ;
        }
        push (c) ;
    }
  }
  while (top != -1) {
      postfix [p++] = pop ();
  }
  postfix [p] = '\0' ;
}
int main () {
  char postfix [MAX] ;
  printf ("Enter a valid parenthesized infix exp: ");
  scanf ("%s", infix) ;
  infixtopostfix (infix, postfix) ;
  printf ("Postfix Expression : %s \n", postfix);
  return 0;
}
```

O/P Enter a valid parenthesized infix exp: ~~a+b~~
a+(b*c-(d/e^f)*g)*h)
Postfix Expression : abc* def ^/- g* +h*

MG
10/10/25