Name :- Siddhi Vinod Pande **Date:-**
Roll No:- 66 **Class:-** SYBCA **Batch:-**
Practical 3 :- Implementation of infix to postfix using stack .

```
#include<stdio.h>
#include<string.h>
class InToPo {
   char *p, *q, *s;
   int top;

public:
   InToPo() {
      p = new char[50];
      q = new char[50];
      s = new char[50];
      top = -1;
   }

   void getInfix() {
      cout << "\nEnter the infix expression: ";
      cin.getline(q, 50);
   }

   void convert() {
      int i = 0, j = 0;

      while (q[i] != '\0') {

         if ((q[i] >= 'a' && q[i] <= 'z') ||
            (q[i] >= 'A' && q[i] <= 'Z') ||
            (q[i] >= '0' && q[i] <= '9')) {
            p[j++] = q[i];
         }

         else if (q[i] == '(') {
            s[++top] = '(';
         }
```

```
        else if (q[i] == '+' || q[i] == '-' || q[i] == '/' || q[i] == '*' || q[i] == '^') {
            while (top != -1 && precedence(s[top]) >= precedence(q[i])) {
                p[j++] = s[top--];
            }
            s[++top] = q[i];
        }

        else if (q[i] == ')') {
            while (top != -1 && s[top] != '(') {
                p[j++] = s[top--];
            }
            top--;
        }
        i++;
    }


    while (top != -1) {
        p[j++] = s[top--];
    }
    p[j] = '\0';

    cout << "\nPostfix is: " << p << endl;
  }

  int precedence(char ch) {
    switch (ch) {
        case '^': return 4;
        case '*':
        case '/': return 3;
        case '+':
        case '-': return 2;
        default: return 1;
    }
  }
};

int main() {
  InToPo obj;
  obj.getInfix();
```

```
    obj.convert();
    return 0;
}
```

OUTPUT

```
Enter the infix expression:--(a+b)-c

Postfix is:--ab+c*

Enter the inf ix expression:--a+b

Postfix is:-- ab+

Enter the infix expression:--a*b+c

Postfix is: --abc+
```