LAB.2

Ajith.M.S
1BM19CS010

```
#include <stdio.h>
# define MAX 100
  char stack [MAX];
  int top = -1;

void push(char ch)
 {
  if (top ==MAX-1)
  printf("Stack is full\n");
  else
  {
   top++;
   stack [top]=ch;
 }}
char pop()
 { char item;
  if(top= =-1)
  printf("\n Stack is empty!");
  else
  {
   item =stack [top];
    top--;
    return item;
 }
}
```

```
int stack empty()
{
  if (top ==-1) return1;
  else return 0;
}

Char stack top()
{
  if (top==-1)
   printf ("\n stack is empty!");
  else
   return stack[top];
}

int priority (char ch)
{
 switch(ch)
 {
  Case '+':
  Case '-': return (1);
  Case '*':
  Case '/': return (2);
  Case '^': return(3);
  default: return(0);
 } }

int main(int argc, char **argv)
{
   char infix[100];
  int i, item;
  printf("enter the infix expression:");
  scanf("%s", infix);
```

```
print("Expression: %s",infix);
printf("\n Postfix : ");
  i=0;
   while(infix[i]!='\0')
 {
  switch(infix[i])
 {
   case '(': push(infix[i]);
             break;
  Cas ')': while(item=pop()!='(')
             print(("%c",item);
            break;
Case '+':
 Case '-':
 Case '*':
Case '/':
Case '^':
               while(!stack empty()&& priority(infix[i]<=
            {                                    priority(stack[o
             item=pop();
               printf("%c",item);
            }
      push(infix[i]);
          break;
  default: printf("%c",infix[i]);
              break;
    }
           i++;
 }
While (!stack empty())
 {
   char item;
   item=pop():;
    printf("%c", item);
  }
  printf("\n");
   return 0;
  }
```