```
#include<stdio.h>
#define MAX 100
char stacks[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 stackEmpty()
{
  if(top==-1) return 1;
  else return 0;
}
char stackTop()
{
  if (top==-1)
   printf("\n Stack is empty!");
  else
    return stack[top];
  }
int priority(char ch)
{
   case '+';
    case '-' : return (1);
    case '*';
    case '/' : return(2);
   default : return(0);
}
int main(int argc, char argv
{
   int infix[100];
   int i, item;
   printf(" enter the infix expression:");
   scanf("%s", infix);
   Printf(" expression: %s", infix);
   Printf("\n postfix : ");
    i=0;
```

```
while (infix[i] != '\0')
{
 switch (infix[i])
 {
   case '(' : push (infix[i]);
              break;
   case ')' : while (item = pop()) != '(')
              printf ("%c", item);
   case '+' ;
   case '-' ;
   case '*' ;
   case '/' ;
              while(! stackempty() && priority
                infix[i]) <= priority(stacktop
              {
                item = pop();
                printf ("%c", item);
              }
              push (infix[i]);
              break;
   default : printf ("%c", infix[i]);
             break;
 }
 i++;
}
while(! stack empty())
 { char item;
   int = pop();
  printf ("%c", item);
 }
```

```
    printf("\n");
    return 0;
}
```