LAB-2

3) WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators +, -, * ./ ?

```
#include <stdio.h>
#include <string.h>
#include <process.h>
int F(Char symbol)
{
  Switch(Symbol)
  {
    Case '+' :
    Case '-' : return 2;
    Case '*' :
    Case '/' : return 4;
    Case '$' :
    Case '(' : return 0;
    Case '#' : return -1;
    default : return 8;
  }
}
int G(Char Symbol)
{
  Switch(Symbol)
  {
    Case '+':
    Case: '-' : return 1;
    Case '*':
    Case '/' : return 3;
Case '^' :
Case '$' : return 6;
Case '(' : return 9;
Case ')' : return 0;
 default : return 7;
}
}
Void infix_postfix(Char infix[], Char postfix[])
{
  int top, i, j;
  Char s[30], Symbol;
   top = -1;
   s[++top] = '#';
    j=0;
   for(i=0; i< strlen(infix); i++)
  {
   Symbol = infix[i];
    while(F(s[top]) > G(Symbol))
     {
      Postfix[j] = s[top--];
       j++;
     }
    if (F(s[top]) != G(Symbol))
       s[++top] = Symbol;
     else
     top --;
   }
   while (s[top] != '#')
    {
      Postfix[j++] = s[top--];
    }
      postfix[j] = '\0';
   }
```

Date
Page

```
Void main()
{
Char infix [20];
Char postfix [20];
Clscr();
printf("Enter the Valid infix expression\n");
Scanf("%s", infix);
infix_postfix(infix, postfix);
printf(" the postfix experrsion is \n");
 printf(" %s\n", postfix);
}
```