Lab-3

Q. WAP to convert a given valid parenthesized infix arthemetic expression to postfix.

Solⁿ
```
# include <stdio.h>
# include (string.h)

int F(char symbol)
{
  switch (symbol)
  {
    Case '+' :
    Case '-' : return 2;
    Case '*' :
    Case '/' : return 4;
    Case '^' :
    case '$' : return 5;
    Case '(' : return 0;
    case '#' : return -1;
    default : return 8;
  }
}
int G(char symbl)
{ switch(symbol)
{
  case '+' :
  case '-' : retur 1;
```

```
i       case '*':
        case '/': return 3;
        case '^':
        case '$': return 6;
        case '(': return 9;
        case ')': return 0;
        default : return 7;
        }
    }
void infix_postfix (char infix[ ], ch
        postfix[ ])
    {
        int top, i, j;
        char s[30], symbl;
        top = -1;
        s[++top] = '#';
        j = 0;
        for (i=0; i<strlen(infix); i++)
        {
            symbl = 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--];
    }
    post fix [j] = '\0';
   }
   void main ()
   {
     Char infix [20];
     Char postfix [20];
   printf ("Enter the valid infix expression\n");
     scanf ("%s", infix);
      infix_postfix (infix, postfix);
      printf ("The postfix exp. is \n");
      printf ("%s\n", postfix);

     }
```