05/10/20

NAME: SAIPRAVEEN MARNI

USN: 1BM19CS138

LAB- PROGRAM: 2

② WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int F(char symbol)
{
  switch (symbol)
  {
    case '+':
    Case '-' : return 2;
    case '*':
    Case '/' : return 4;
    Case '^':
    case '$' : return 5;
    case '(' : return 0;
    case '#' : return -1;
   default : return 8;
  }
}
int G(char symbol)
{
  switch(symbol)
  {
    case '+':
    Case '-' : return 1;
    Case '*' :
    case '/' : return 3;
    case '^':
   Case '$' : 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' ;
}
  void main ()
{
  char infix [20];
  char postfix [20];
printf ("Enter the valid infix expression\n");
```

```
    scanf("%s", infix);
    infix_postfix(infix, postfix);
  printf("the positive exp is \n");
    printf("%s \n", postfix);
     getch();
}
```