LP:02 (week 3) [Infix+ Postfix Prefix expression.]

```
# 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 '$': return 5;
     case '(': return 0;
      case '#': return -1;
      ~~case~~ *
      default : return 8;
     }
 }
   int G (char Symbol)
  {
    Switch (Symbol)
      {
         case '+':
         case '-': return 1;
         case '*':
         case '/': return 3;
         case '^':
         case '$': return 6;
         case '(': return 9;
          case ')': return 0;
          ~~case~~
          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];
    clrscr();
```

Scanned by CamScanner

```
printf ("enter the valid infix expression\n");
 Scanf (" %s", infix);
 infix_postfix (infix, postfix);
 printf (" the postfix exp is \n");
 printf ("%s \n", postfix);
 getch ();
}
```

1) Conversion from infix to postfix

(a) ((A+(B-C) * D) ^ E+F)

→ ABC-D*+ E^F+

(b) x^y^z-M+N+P/Q

→ xyz^^ M-N+PQ/+

(c) ((a+b) * c-(d-e))^(f+g)

→ ab+c*de--fg+^

(d) (A+(B-C) *D)

→ ABC-D*+

(e) a^b*c-d+e/f/(g+h)

→ ab^c*d-ef/gh+/+