Lab program - 2

```
#include <stdio.h>
# include < string.h>
int main()
{
    char stack[20], iexp[50], pexp[50];
    int i, p=0, s=-1, j, Z=1, operands =0, operators =0,
    obracket =0, cbracket=0;
    printf("enter the expression\n");
    scanf(" %s ", iexp);
    for (i= 0; i< strlen(iexp); i++)
    {
     if (iexp[i]=='+'||iexp[i]=='-'||iexp[i]=='*'||iexp[i]
      =='/'||iexp[i]=='+'||iexp[i]=='^')
     operators ++;
     if((iexp[i]>65 && iexp[i]<=90)|| (iexp[i] >=97 && iexp[i]<=
          122))
     operands ++;
      if (iexp[i]=='(')
      obracket++;
      if (iexp[i] ==')')
      cbracket++;
    }
     if (operands != (operators+1) || Obracket != cbracket)
     Z=0;
    for (i=0; i< strlen(iexp); i++)
    {
      if ((iexp[i]>65 && iexp[i]<=90)|| (iexp[i]>=97 &&
         iexp[i]<=122))
     {   pexp[p]=iexp[i];
          p++;
     }
```

```
else if ( iexp[i]== '(')
    {   s++;
        stack[s]=iexp[i];
    }
  else  if (iexp[i] ==')')
   {
      if (iexp[i-1]=='+'|| iexp[i-1]=='-'|| iexp[i-1]=='*'
        || iexp[i-1]=='/' || iexp [i-1]== '+')
      z=0;
    do
  {     if (stack[s]=='(')
        {    s--;
          break;
        }
        pexp[p]= stack[s];
        p++;
        s--;
    } while ('c'=='c');
 }
else  if (iexp[i]=='+' ||iexp [i] =='-')
   {
      if (stack[s] =='(' ||s==-1)
      {  s++;
         stack[s] = iexp[i];
      }
      else
    {    do
         {
            f=0;
            if (stack[s] =='(')
             {   s++;
            stack[s]= iexp [i];
            f=1;
            break;
        }
      pexp[p]= stack[s];
      p++;
    s--;
   }
   while (s!=-1)
   if (f==0)
   {  s++;
      stack[s] = iexp[i];
   }
  }
}
 else if (iexp[i] == '*' || iexp[i] == '/')
 {   if (stack[s]== '(' || stack[s] == '+' || stack[s] == '-'
             || s==-1)
     {  s++;
        stack[s]= iexp[i];
     }
    else
    {  do {
             f=0;
           if (stack[s] == '+' || stack[s] == '-' || stack[s]=='(')
           {   s++;
               stack[s]= iexp[i];
               f=1;
               break;
           }
           pexp[p]= stack[s];
         p++;  s--;
     }   while (s!=-1);
```

```
if (f==0)
{  s++;
   stack[s]=iexp[i];
  }}}
 else if (iexp[i]=='^')
   { if (stack[s]=='(' || stack[s] == '+' ||
   stack[s]=='-' || s==-1 || stack[s]=='*'
    || stack[s]=='/')
   {      s++;
        stack[s]=iexp[i];
   }
   else
   {  do {
            f=0;
          if (stack[s]=='+'|| stack[s] == '-'||
        stack[s]=='('|| stack[s]=='*'|| stack[s]=='/')
       (  s++;
          stack[s]=iexp[i];
          f=1;
          break;
      }  pexp[p]=stack[s];
          p++;
        s--;
   } while (s!=-1)
    if (f==0)
     {  s++;
      stack[s]=iexp[i];
    }}}}
if (s!=-1)
{     do {
           pexp[s]=stack[s],
             p++; s--;
       }
```

```
        while (s!=-1);
    }
  if (z==0)
  printf(" Invalid expression \n");
  else
  { printf(" The postfix expression is:\n");
    for (i=0; i<p; i++)
    {  printf("%c", pexp[i]);
    }
  } return 0;
}
```