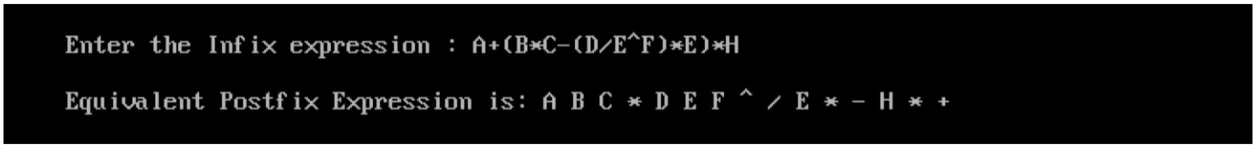
```
/*Write a program that uses stack operations to convert a
given infix expression into its postfix equivalent*/
#include<stdio.h>
#include<conio.h>

char stack[100];
int top = -1;

void push(char x)
{
    stack[++top] = x;
}

char pop()
{
    if(top == -1)
 return -1;
    else
 return stack[top--];
}

int priority(char x)
{
    if(x == ’(’||x==’)’)
 return 0;
    if(x == ’+’ || x == ’-’)
 return 1;
    if(x == ’*’ || x == ’/’)
 return 2;
    if(x==’^’)
 return 3;
    return 0;
}
void main()
{
    char exp[100];
    char *e, x;
    clrscr();
    printf("\nEnter the Infix expression : ");
    scanf("%s",exp);
    printf("\n");
    e = exp;
    printf("Equivalent Postfix Expression is: ");
    while(*e != ’\0’)
    {
 if(isalnum(*e))
     printf("%c ",*e);
 else if(*e == ’(’)
     push(*e);
 else if(*e == ’)’)
 {
     while((x = pop()) != ’(’)
  printf("%c ", x);
 }
 else
 {
     while(priority(stack[top]) >= priority(*e))
  printf("%c ",pop());
     push(*e);
 }
 e++;
    }
    while(top != -1)
    {
 printf("%c ",pop());
    }
 getch();
}
```

Output Program 5



```
Enter the Infix expression : A+(B*C-(D/E^F)*E)*H

Equivalent Postfix Expression is: A B C * D E F ^ / E * - H * +
```