```
#include<stdio.h>
#include<ctype.h>

char Stack[25]; //Operator stack
int top=-1;

void push(char ele)
{
        Stack[++top] = ele;
}

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

int stack_priority(char x)
{
  if(x== '#')
          return -1;
  if(x== '(')
          return 0;
  if(x=='+'||x=='-')
           return 2;
  if(x=='*'||x=='/'||x=='%')
           return 4;
         if(x=='^')
           return 5;
```

```
}

int sym_priority(char x)
{
	if(x== '(')
		return 9;
	if(x=='+'||x=='-')
		return 1;
	if(x=='*'||x=='/'||x=='%')
		return 3;
	if(x=='^')
		return 6;
}

void main()
{
 char Infix[20],sym,x;
 int i;
 printf("\n\n\t\tEnter the Valid Infix Expression: ");
 scanf("%s",Infix);
 i=0;
 push('#');
 printf("\n The postfix expression is :");

    while(Infix[i] != '\0')
  {
		sym=Infix[i];
	if(isalnum(sym))
		printf("%c",Infix[i]);
	else if(sym=='(')
		push(sym);
```

```
        else if(sym==')')
          {
              while((x=pop()) != '(')
                        printf("%c",x);
          }
         else
          {
            while((stack_priority(Stack[top]))>=sym_priority(sym))
              printf("%c",pop());
            push(sym);
          }
        i++;
   }  //end of while

   while(top != 0)
    printf("%c",pop());
}
```