Program:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#define max 100
char stack[max];
int top=-1;
void InfixToPostfix(char source[],char target[]);
void push(char st[],char a);
char pop(char st[]);
int PriorityOrder(char op);
bool isDigit(char a);
bool isAlpha(char c);
        int main()
        {
                char infix[100],postfix[100];
                printf("Enter infix expression \n");
                gets(infix);
                strcpy(postfix,"");
                InfixToPostfix(infix,postfix);
                return 0;
        }
        void InfixToPostfix(char source[],char target[])
        {
                 int i=0,j=0;
                 char temp;
                 strcpy(target,"");
                while(source[i]!='\0')
                {
                        if(source[i]=='(')
                        {
```

```
push(stack,source[i]);
i++;
}

else if (source[i]==')')
{
        while(top!=-1 && (stack[top]!='(') )
        {
        target[j]=pop(stack);
        j++;
        }
        if(top==-1)
        {
        printf("Incorrect expression \n");
        }
        temp=pop(stack);
        i++;
}
else if(isDigit(source[i]) || isAlpha(source[i]))
{
        target[j]=source[i];
        j++;
        i++;
}

else if(source[i]=='+'|| source[i]=='-'|| source[i]=='*'||
source[i]=='/'||source[i]=='^'||source[i]=='$)
{
        while(top!=-1 && (stack[top]!='(') &&
        (PriorityOrder(stack[top])>=PriorityOrder(source[i])) )
        {
```

```
                                target[j]=pop(stack);
                                j++;
                        }
                        push(stack,source[i]);
                        i++;
                }
                else
                printf("Incorrect expression \n");
        }

        while((top!=-1) && (stack[top]!= '('))
        {
        target[j]=pop(stack);
        j++;
        }
        target[j]='\0';
        printf("Your postfix expression is \n");
        puts(target);
}

void push(char st[],char a)
{
         if(top==max-1)
                printf("Stack overflow");
        else
                {
                top++;
                st[top]=a;
                }
}
```

```
char pop(char st[])
{
	char x=' ';
	if(top==-1)
		printf("Stack Underflow \n");
	else
	{
		x=stack[top];
		top--;
	}
	return x;
}

int PriorityOrder(char op)
{
	if(op=='^' || op=='$')
		return 3;
	else if(op=='*' || op=='/')
		return 2;
	else if(op=='+' || op=='-')
		return 1;
}



bool isDigit(char a)
{
	 if(a>='0' && a<='9')
		return true;
	 else
		return false;
}
```

```
bool isAlpha(char c)
{
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))
                return true;
         else
                return false;
}
```

Output:

```
Enter infix expression
(A+B)*(C-D)$E*F
Your postfix expression is
AB+CD-E$*F*
```