

```

#include<stdio.h>

#include<string.h>

char stack[50];

int top=-1;

void post(char infix[]);

void push(char);

char pop();


void main()
{
    char infix[25];

    printf("\nENTER THE INFIX EXPRESSION = ");

    gets(infix);

    post(infix);

    getch();
}


void push(char symb)
{
    if(top>=49)
    {
        printf("\nSTACK OVERFLOW");

        getch();

        return;
    }

    else

    {
        top=top+1;

        stack[top]=symb;
    }
}

```

```

char pop()
{
    char item;
    if(top== -1)
    {
        printf("\nSTACK IS EMPTY");
        getch();
        return(0);
    }
    else
    {
        item=stack[top];
        top--;
    }
    return(item);
}

int preced(char ch)
{
    if(ch==47)
    {
        return(5);
    }
    else if(ch==42)
    {
        return(4);
    }
    else if(ch==43)
    {
        return(3);
    }
    else

```

```

        return(2);
    }
void post(char infix[])
{
    int l;
    int index=0,pos=0;
    char symbol,temp;
    char postfix[40];
    l=strlen(infix);
    push('#');
    while(index<l)
    {
        symbol=infix[index];
        switch(symbol)
        {
            case '(': push(symbol);
            break;
            case ')': temp=pop();
            while(temp!='(')
            {
                postfix[pos]=temp;
                pos++;
                temp=pop();
            }
            break;
            case '+':
            case '-':
            case '*':
            case '/':
            case '^':
                while(preced(stack[top])>=preced(symbol))

```

```

        {
            temp=pop();
            postfix[pos]=temp;
            pos++;
        }
        push(symbol);
        break;
    default: postfix[pos++]=symbol;
        break;
    }
    index++;
}
while(top>0)
{
    temp=pop();
    postfix[pos++]=temp;
}
postfix[pos++]='\0';
puts(postfix);
return;

```

```

}
ENTER THE INFIX EXPRESSION = (((a*b)+(c/d))-(e*f))
STACK IS EMPTY

```