

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#define MAX 20
void InfixtoPostfix(char source[],char target[]);
int getpriority(char);
void push(char [],char);
char pop(char []);
char st[MAX];
int top=-1;
int main()
{
    char infix[20],postfix[20];
    clrscr();
    printf("\n Enter any infix expression");
    gets(infix);
    InfixtoPostfix(infix,postfix);
    printf("\n The Corresponding postfix expression is :");
    puts(postfix);
    getch();
    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(st,source[i]);
            i++;
        }
        else if(source[i] ==')')
        {
            while((top !=-1) && (st[top] !='('))
            {
                target[j]=pop(st);
                j++;
            }
            if(top== -1)
            {
                printf("\n Incorrect Expression");
                exit(1);
            }
            temp=pop(st);
            i++;
        }
    }
    target[j]=temp;
    j++;
    target[j]='\0';
    printf("\n The postfix expression is :");
    puts(target);
    getch();
    return 0;
}

```

```

    }
    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] == '%')
    {
        while((top != -1) && (st[top] != '(') && getpriority(st[top]) >=
getpriority(source[i]))
        {
            target[j]=pop(st);
            j++;
        }
        push(st,source[i]);
        i++;
    }
    else
    {
        printf("\n Incorrect Expression");
        exit(1);
    }
}
while((top != -1) && (st[top] != '('))
{
    target[j]=pop(st);
    j++;
}
target[j]='\0';
}
int getpriority(char op)
{
    if(op == '/' || op == '*' || op == '%')
        return 1;
    else if(op == '+' || op == '-')
        return 0;
}
void push(char st[],char val)
{
    if(top==MAX-1)
        printf("\n Stack Overflow");
    else
    {
        top++;
        st[top]=val;
    }
}
char pop(char st[])

```

```
{
    char val=' ';
    if(top==-1)
        printf("\n Stak Underflow");
    else
    {
        val=st[top];
        top--;
    }
    return val;
}
```