

```

#define SIZE 50 /* Size of Stack */
#include <ctype.h>
#include <stdio.h>
char s[SIZE];
int top = -1; /* Global declarations */
push(char elem) /* Function for PUSH operation */
{
    s[++top] = elem;
}
char pop() /* Function for POP operation */
{
    return (s[top--]);
}
int pr(char elem) /* Function for precedence */
{
    switch (elem)
    {
        case '#': return 0;
        case '(': return 1;
        case '+': case '-': return 2;
        case '*': case '/': case '%': return 3;
        case '^': return 4;
    }
}
void main() /* Main Program */
{
    char infix[50], postfix[50], ch, elem;
    int i = 0, k = 0;
    printf("\n\nRead the Infix Expression ? ");
    scanf("%s", infix);
    push('#');
    while ((ch = infix[i++]) != '\0')
    {
        if (ch == '(') push(ch);
        else if (isdigit(ch)) postfix[k++] = ch;
        else if (ch == ')')
        {
            while (s[top] != '(')
                postfix[k++] = pop();
            elem = pop(); /* Remove ( */
        }
        else /* Operator */
        {
            while (pr(s[top]) >= pr(ch))
                postfix[k++] = pop();
        }
    }
}

```

```
        push(ch);
    }
}
while (s[top] != '#') /* Pop from stack till empty */
    pofx[k++] = pop();
pofx[k] = '\0'; /* Make pofx as valid string */
printf("\n\nGiven Infix Expn: %s Postfix Expn: %s\n", infix,
pofx);
}
```