Conversion of Infix arithmetic expression to Postfix expression.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

int ind = 0, pos = 0, top = -1, length;
char symbol, temp, infix[20], postfix[20], stack[20];

void infixtopostfix();
void push(char symbol);
char pop();
int pred(char symbol);

void main()
{
   printf("Enter infix expression\n");
   scanf("%s", infix);

   infixtopostfix();

   printf("Infix expression: %s\n", infix);
   printf("Postfix expression: %s\n", postfix);
}

void infixtopostfix()
{
   length = strlen(infix);
   push('#');
   while (ind < length)
```

```
{
    symbol = infix[ind];
    switch (symbol)
    {
    case '(':
        push(symbol);
        break;
    case ')':
        temp = pop();
        while (temp != '(')
        {
            postfix[pos] = temp;
            pos++;
            temp = pop();
        }
        break;
    case '+':
    case '-':
    case '*':
    case '/':
    case '^':
        while (pred(stack[top]) >= pred(symbol))
        {
            temp = pop();
            postfix[pos++] = temp;
        }
        push(symbol);
        break;
    default:
        postfix[pos++] = symbol;
```

```
        }
        ind++;
    }
    while (top > 0)
    {
        temp = pop();
        postfix[pos++] = temp;
    }
}

void push(char symbol)
{
    top = top + 1;
    stack[top] = symbol;
}

char pop()
{
    char symb;
    symb = stack[top];
    top = top - 1;
    return (symb);
}

int pred(char symbol)
{
    int p;
    switch (symbol)
    {
    case '^':
```

```
        p = 3;
        break;
    case '*':
    case '/':
        p = 2;
        break;
    case '+':
    case '-':
        p = 1;
        break;
    case '(':
        p = 0;
        break;
    case '#':
        p = -1;
        break;
    }
    return (p);
}
```

OUTPUT: