

```
#include<studio.h>
```

```
#include <ctype.h>
```

```
#define MAX 100
```

```
char stack[MAX];
```

```
int top = -1;
```

```
void push(char x) {
```

```
    stack[++top] = x;
```

```
}
```

```
char pop() {
```

```
    if(top == -1)
```

```
        return -1;
```

```
    else
```

```
        return stack[top--];
```

```
}
```

```
int priority(char x) {
```

```
    if(x == '(')
```

```
        return 0;
```

```
    if(x == '+' || x == '-')
```

```
        return 1;
```

```
    if(x == '*' || x == '/')
```

```
        return 2;
```

```
    return 0;
```

```
}
```

```
int main() {
```

```
    char exp[MAX];
```

```
    char *e, x;
```

```
    printf("Enter Infix Expression: ");
```

```
    scanf("%s", exp);
```

```
    printf("Postfix Expression: ");
```

```
    e = exp;
```

```
    while(*e != '\0') {
```

```
        if(isalnum(*e))
```

```
            printf("%c", *e);
```

```
        else if(*e == '(')
```

```
            push(*e);
```

```
        else if(*e == ')') {
```

```
            while((x = pop()) != '(')
```

```
                printf("%c", x);
```

```
        }
```

```
        else {
```

```
            while(priority(stack[top]) >= priority(*e))
```

```
                printf("%c", pop());
```

```
            push(*e);
```

```
        }
```

```
        e++;  
    }  
  
    while(top != -1)  
        printf("%c", pop());  
  
    return 0;  
}
```