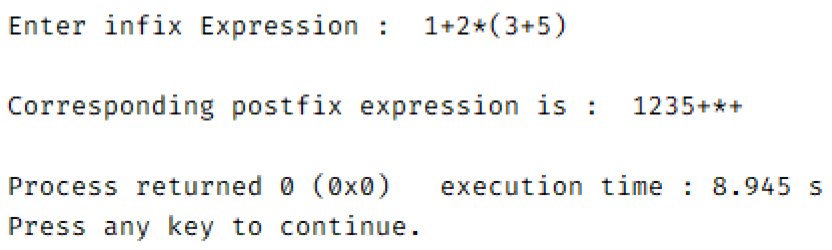
Infix to postfix:

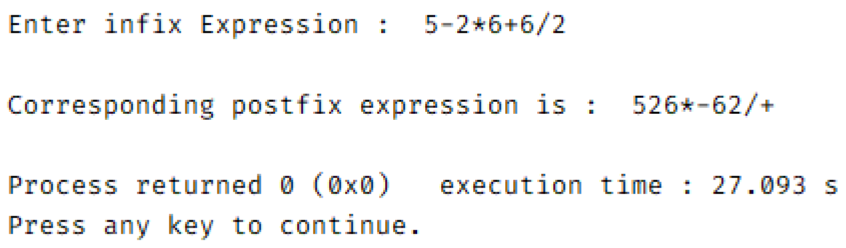
```
//code
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define MAXSTACK 100
#define SIZE 100

char stack[MAXSTACK];
int top = -1;

void push(char item) {
    if (top >= MAXSTACK-1) {
        printf("OVERFLOW");
        return;
    } else {
        top++;
        stack[top] = item;
    }
}
```

```
char pop() {
    char item;
    if (top < 0) {
        printf("UNDERFLOW");
    } else {
        item = stack[top];
        top--;
        return item;
    }
}

int isOperator(char symbol) {
    if (symbol=='+' || symbol=='-' || symbol=='*' || symbol=='/' || symbol=='^') {
        return 1;
    } return 0;
}

int precedence(char symbol) {
    if(symbol == '^') {
        return 3;
    } else if(symbol == '/' || symbol == '*') {
        return 2;
    } else if(symbol == '+' || symbol == '-') {

        return 1;

    } else {

        return 0;

    }

}


void infixToPostfix(char infix[], char postfix[]) {

    int i=0, j=0;

    char item, x;

    strcat(infix, ")");

    push('(');

    for(i=0 ; infix[i] != '\0' ; i++) {

        item = infix[i];

        if(item == '(') {

            push('(');

        } else if(isdigit(item)) {

            postfix[j++] = item;

        } else if(isOperator(item)) {

            x = pop();

            while(isOperator(x)==1 && precedence(x)>=precedence(item)) {

                postfix[j++] = x;

                x=pop();
            }
            push(x);
            push(item);
        } else if(item == ')') {
            x = pop();
            while (x != '(') {
                postfix[j++] = x;
                x = pop();
            }
        }
    }
    if(top > 0) {
        printf("Invalid expression");
    }
    postfix[j]='\0';
}

int main() {
    char infix[SIZE], postfix[SIZE];
    printf("Enter infix Expression :  ");
    gets(infix);
    infixToPostfix(infix, postfix);
    printf("\nCorresponding postfix expression is :  ");
    puts(postfix);

}
```

//output



```
Enter infix Expression :  1+2*(3+5)

Corresponding postfix expression is :  1235+*+

Process returned 0 (0x0)   execution time : 8.945 s
Press any key to continue.
```



```
Enter infix Expression :  5-2*6+6/2

Corresponding postfix expression is :  526*-62/+

Process returned 0 (0x0)   execution time : 27.093 s
Press any key to continue.
```