

EXPERIMENT-3

Program :

```
#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdlib.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 precedence(char x) {
    if (x == '(') return 0;
    if (x == '+' || x == '-') return 1;
    if (x == '*' || x == '/') return 2;
    return 0;
}

void infixToPostfix(char infix[], char postfix[]) {
    char x;
```

```

int i = 0, k = 0;
while (infix[i]  $\neq$  '\0') {
    if (isalnum(infix[i])) {
        postfix[k++] = infix[i]; // If operand, add to output
    }
    else if (infix[i] == '(') {
        push(infix[i]);
    }
    else if (infix[i] == ')') {
        while ((x = pop())  $\neq$  '(') {
            postfix[k++] = x;
        }
    }
    else { // operator
        while (precedence(stack[top])  $\geq$  precedence(infix[i])) {
            postfix[k++] = pop();
        }
        push(infix[i]);
    }
    i++;
}

while (top  $\neq$  -1) {
    postfix[k++] = pop();
}

postfix[k] = '\0';
}

int evalStack[MAX];

```

```
int evalTop = -1;

void evalPush(int x) {
    evalStack[++evalTop] = x;
}

int evalPop() {
    return evalStack[evalTop--];
}

int evaluatePostfix(char postfix[]) {
    int i = 0;
    int a, b;
    while (postfix[i] != '\0') {
        if (isdigit(postfix[i])) {
            evalPush(postfix[i] - '0');
        }
        else {
            b = evalPop();
            a = evalPop();
            switch (postfix[i]) {
                case '+': evalPush(a + b); break;
                case '-': evalPush(a - b); break;
                case '*': evalPush(a * b); break;
                case '/': evalPush(a / b); break;
            }
        }
        i++;
    }
    return evalPop();
}
```

```
}  
  
int main() {  
    char infix[MAX], postfix[MAX];  
    printf("Enter Infix Expression: ");  
    scanf("%s", infix);  
    infixToPostfix(infix, postfix);  
    printf("Postfix Expression: %s\n", postfix);  
    int result = evaluatePostfix(postfix);  
    printf("Evaluated Result: %d\n", result);  
    return 0;  
}
```

Output:

```
cseb2@sjcet-OptiPlex-SFF-7020:~/Alwin$ ./a.out  
Enter Infix Expression: 200*(15-16)^6  
Postfix Expression: 2001516-*6^  
Evaluated Result: 1
```