Exp. No. 20
Write a C program to compute TRAILING( ) – operator precedence
parser for the given grammar
E → E + T | T
T → T * F | F
F → (E) | id
Program:

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

// Parsing Table
char arr[18][3] = {
    {'E', '+', 'F'}, {'E', '*', 'F'}, {'E', '(', 'F'}, {'E', ')', 'F'}, {'E', 'i', 'F'},
{'E', '$', 'F'},
    {'F', '+', 'F'}, {'F', '*', 'F'}, {'F', '(', 'F'}, {'F', ')', 'F'}, {'F', 'i', 'F'}, {'F',
'$', 'F'},
    {'T', '+', 'F'}, {'T', '*', 'F'}, {'T', '(', 'F'}, {'T', ')', 'F'}, {'T', 'i', 'F'},
{'T', '$', 'F'}
};

// Production Rules
char prod[6] = "EETTFF";
char res[6][3] = {
    {'E', '+', 'T'}, {'T', '\0', '\0'}, {'T', '*', 'F'}, {'F', '\0', '\0'},
    {'(', 'E', ')'}, {'i', '\0', '\0'}
};

// Stack for Productions
#define STACK_SIZE 10
char stack[STACK_SIZE][2];
int top = -1;

// Function to Install a Rule
void install(char pro, char re) {
    if (top >= STACK_SIZE - 1) {
        printf("Stack Overflow! Too many rules.\n");
        return;
    }
    for (int i = 0; i < 18; ++i) {
```

```
        if (arr[i][0] == pro && arr[i][1] == re) {
            arr[i][2] = 'T';
            stack[++top][0] = pro;
            stack[top][1] = re;
            break;
        }
    }
}

int main() {
    int i, j;
    char pro, re, pri = ' ';

    // Fill Parsing Table
    for (i = 0; i < 6; ++i) {
        for (j = 2; j >= 0; --j) {
            if (strchr("+-()*i$", res[i][j])) {
                install(prod[i], res[i][j]);
                break;
            } else if (strchr("EFT", res[i][j]) && j > 0 && strchr("+-
()*i$", res[i][j - 1])) {
                install(prod[i], res[i][j - 1]);
                break;
            }
        }
    }

    // Processing Stack
    while (top >= 0) {
        pro = stack[top][0];
        re = stack[top--][1];
        for (i = 0; i < 6; ++i) {
            if (res[i][0] == pro && res[i][0] != prod[i]) {
                install(prod[i], re);
            }
        }
    }

    // Print Parsing Table
    printf("\nParsing Table:\n");
```

```
    for (i = 0; i < 18; ++i) {
        printf("\t");
        for (j = 0; j < 3; ++j) {
            printf("%c\t", arr[i][j]);
        }
        printf("\n");
    }

    // Print Production Rules
    printf("\nProductions:\n");
    for (i = 0; i < 18; ++i) {
        if (pri != arr[i][0]) {
            pri = arr[i][0];
            printf("\n\t%c -> ", pri);
        }
        if (arr[i][2] == 'T') {
            printf("%c ", arr[i][1]);
        }
    }

    printf("\n");
    return 0;
}
```