VIT CHENNAI

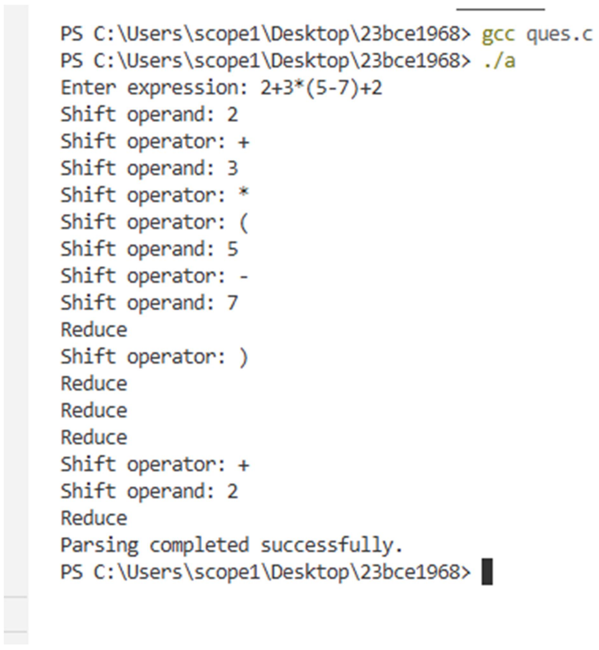
LAB ASSIGNMENT - 9

COMPILER DESIGN LAB – BCSE307P

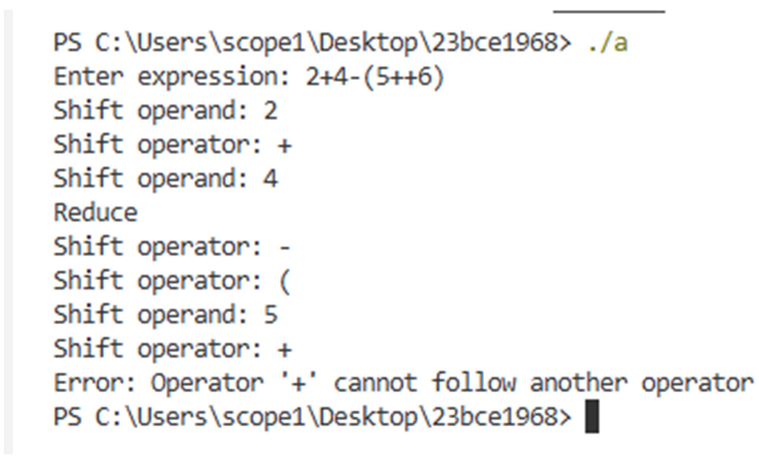
NAME – ADITYARAJ SHRIVASTAVA
REG. NO.- 23BCE1968

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

#define MAX 100

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

char precedenceTable[7][7] = {
    {'>', '>', '<', '<', '<', '>', '>'},
    {'>', '>', '<', '<', '<', '>', '>'},
    {'>', '>', '>', '>', '<', '>', '>'},
    {'>', '>', '>', '>', '<', '>', '>'},
    {'<', '<', '<', '<', '<', '=', '0'},
    {'>', '>', '>', '>', '0', '>', '>'},
    {'<', '<', '<', '<', '<', '0', '='}
};

char terminals[] = {'+', '-', '*', '/', '(', ')', '#'};

int getIndex(char c) {
    for (int i = 0; i < 7; i++) {
        if (terminals[i] == c) return i;
    }
    return -1;
}

char getPrecedence(char a, char b) {
    int i = getIndex(a);
    int j = getIndex(b);
    if (i == -1 || j == -1) return '0';
    return precedenceTable[i][j];
}

void push(char c) {
    if (top < MAX - 1) stack[++top] = c;
}

char pop() {
    if (top >= 0) return stack[top--];
    return '\0';
}

char stackTop() {
    if (top >= 0) return stack[top];
    return '\0';
}

int isOperator(char c) {
    for (int i = 0; i < 7; i++) {
        if (terminals[i] == c) return 1;
    }
    return 0;
}

void reduce() {
    if (stackTop() == ')') {
        pop(); // pop ')'
        while (stackTop() != '(' && top >= 0) {
            pop();
        }
        if (stackTop() == '(') {
            pop(); // pop '('
        } else {
            printf("Error: Mismatched parenthesis during reduce\n");
            return;
        }
        printf("Reduce\n");
    } else {
        pop();
        printf("Reduce\n");
    }
}

void operatorPrecedenceParsing(char* expr) {
    int i = 0;
    char a, b;
    int lastTokenWasOperand = 0;
    push('#');
    while (expr[i] != '\0') {
        if (isdigit(expr[i])) {
            printf("Shift operand: %c\n", expr[i]);
            lastTokenWasOperand = 1;
            i++;
        } else if (isOperator(expr[i]) || expr[i] == '(' || expr[i] == ')') {
            if (!lastTokenWasOperand && expr[i] != '(') {
                printf("Error: Operator '%c' cannot follow another operator\n",
expr[i]);
                return;
            }
            a = stackTop();
            b = expr[i];
            char prec = getPrecedence(a, b);
            if (prec == '<' || prec == '=') {
                printf("Shift operator: %c\n", expr[i]);
                push(expr[i]);
                lastTokenWasOperand = (expr[i] == ')') ? 1 : 0;
                i++;
            } else if (prec == '>') {
                reduce();
                lastTokenWasOperand = 1;
            } else {
                printf("Error: Invalid precedence between '%c' and '%c'\n", a,
b);
                return;
            }
        } else {
            printf("Error: Invalid character '%c' in input\n", expr[i]);
            return;
        }
    }
    if (!lastTokenWasOperand) {
        printf("Error: Expression ends with an operator\n");
        return;
    }
    while (stackTop() != '#') {
        if (stackTop() == '(') {
            printf("Error: Mismatched parenthesis\n");
            return;
        }
        reduce();
    }
    printf("Parsing completed successfully.\n");
}

int main() {
    char expr[MAX];
    printf("Enter expression: ");
    scanf("%s", expr);
    operatorPrecedenceParsing(expr);
    return 0;
}
```

OUTPUT:



```
PS C:\Users\scope1\Desktop\23bce1968> gcc ques.c
PS C:\Users\scope1\Desktop\23bce1968> ./a
Enter expression: 2+3*(5-7)+2
Shift operand: 2
Shift operator: +
Shift operand: 3
Shift operator: *
Shift operator: (
Shift operand: 5
Shift operator: -
Shift operand: 7
Reduce
Shift operator: )
Reduce
Reduce
Reduce
Shift operator: +
Shift operand: 2
Reduce
Parsing completed successfully.
PS C:\Users\scope1\Desktop\23bce1968>
```



```
PS C:\Users\scope1\Desktop\23bce1968> ./a
Enter expression: 2+4-(5++6)
Shift operand: 2
Shift operator: +
Shift operand: 4
Reduce
Shift operator: -
Shift operator: (
Shift operand: 5
Shift operator: +
Error: Operator '+' cannot follow another operator
PS C:\Users\scope1\Desktop\23bce1968>
```