LAB ASSIGNMENT – 10

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][10];
int top = -1;
int tempCount = 1;

void push(char *str) {
    strcpy(stack[++top], str);
}

char* pop() {
    return stack[top--];
}

void newTemp(char *temp) {
    sprintf(temp, "t%d", tempCount++);
}

int isOperator(char c) {
    return (c == '+' || c == '-' || c == '*' || c == '/');
}

int precedence(char c) {
    if (c == '*' || c == '/')
        return 2;
    else if (c == '+' || c == '-')
        return 1;
    else
        return 0;
}

void infixToPostfix(char* infix, char* postfix) {
    char stackOp[MAX];
    int topOp = -1;
    int i = 0, j = 0;

    while (infix[i] != '\0') {
        if (isalnum(infix[i])) {
            postfix[j++] = infix[i];
        }
        else if (infix[i] == '(') {
```

```
            stackOp[++topOp] = '(';
        }
        else if (infix[i] == ')') {
            while (topOp != -1 && stackOp[topOp] != '(') {
                postfix[j++] = stackOp[topOp--];
            }
            topOp--;
        }
        else if (isOperator(infix[i])) {
            while (topOp != -1 && precedence(stackOp[topOp]) >=
precedence(infix[i])) {
                postfix[j++] = stackOp[topOp--];
            }
            stackOp[++topOp] = infix[i];
        }
        i++;
    }

    while (topOp != -1) {
        postfix[j++] = stackOp[topOp--];
    }
    postfix[j] = '\0';
}

void generateTAC(char *postfix) {
    int i = 0;
    char temp1[10], temp2[10], temp[10];

    while (postfix[i] != '\0') {
        if (isalnum(postfix[i])) {
            char operand[2];
            operand[0] = postfix[i];
            operand[1] = '\0';
            push(operand);
        }
        else if (isOperator(postfix[i])) {
            char *op2 = pop();
            char *op1 = pop();
            newTemp(temp);
            printf("%s = %s %c %s\n", temp, op1, postfix[i], op2);
            push(temp);
        }
        i++;
    }
}
```

```c
int main() {
    char input[100];
    printf("Enter expression:  ");
    scanf("%s", input);

    char lhs = input[0];
    char rhs[100];

    strcpy(rhs, &input[2]);

    char postfix[100];
    infixToPostfix(rhs, postfix);

    printf("\nPostfix: %s\n", postfix);

    printf("\nThree Address Code:\n");
    generateTAC(postfix);

    char *result = pop();
    printf("%c = %s\n", lhs, result);

    return 0;
}
```

OUTPUT:

1:)

```
PS C:\Users\scope1\Desktop\23bce1968> gcc lab10.c
PS C:\Users\scope1\Desktop\23bce1968> ./a
Enter expression:  x=(a+b)*c

Postfix: ab+c*

Three Address Code:
t1 = a + b
t2 = t1 * c
x = t2
```

2:)

```
PS C:\Users\scope1\Desktop\23bce1968> ./a
Enter expression:  z=p-q/r

Postfix: pqr/-

Three Address Code:
t1 = q / r
t2 = p - t1
z = t2
PS C:\Users\scope1\Desktop\23bce1968>
```