## SAYAN PRAMANICK

## RA1811030010023

# **Lab session 9: Finding LEADING of a variable**

Code – (Turbo C++)

#include<conio.h>

#include<stdio.h>

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'}};

char prod[6] = "EETTFF";

char res[6][3] ={ {'E', '+', 'T'}, {'T', '\0'}, {'T', '\*', 'F'},  {'F', '\0'}, {'(', 'E', ')'}, {'i', '\0'}};

char stack [5][2];

int top = -1;

void install(char pro, char re) {

    int i;

    for (i = 0; i < 18; ++i) {

        if (arr[i][0] == pro && arr[i][1] == re) {

            arr[i][2] = 'T';

            break;

        }

    }

    ++top;

    stack[top][0] = pro;

    stack[top][1] = re;

}

void main() {

    int i = 0, j;

    char pro, re, pri = ' ';

    clrscr();

    for (i = 0; i < 6; ++i) {

        for (j = 0; j < 3 && res[i][j] != '\0'; ++j) {

            if (res[i][j] == '+' || res[i][j] == '\*' || res[i][j] == '(' || res[i][j] == ')' || res[i][j] == 'i' || res[i][j] == '$') {

                install(prod[i], res[i][j]);

                break;

            }

        }

    }

    while (top >= 0) {

        pro = stack[top][0];

        re = stack[top][1];

        --top;

        for (i = 0; i < 6; ++i) {

            if (res[i][0] == pro && res[i][0] != prod[i]) {

                install(prod[i], re);

            }

        }

    }

    for (i = 0; i < 18; ++i) {

        printf("\n\t");

        for (j = 0; j < 3; ++j)

            printf("%c\t", arr[i][j]);

    }

    getch();

    clrscr();

    printf("\n\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]);

    }

    getch();

}

Input –

E+F

E\*F

E(F

E)F

EiF

E$F

F+E

F\*F

F(F

F)F

FiF

F$F

T+F

Output –

Graphical user interface

Description automatically generated

E -> + \* ( i

F -> ( i

T -> \* ( i