

Name :- Shubham Gond

Roll No :- 20cs01062

1)

non terminals: E, T, F, G
terminals: +, -, *, (,), /

```
struct node {
    int isOp;
    int value;
    struct node *left;
    struct node *right;
}
```

Start: $G \rightarrow E$ { $E.op = E.op$, $G.val = E.val$,
 $G.left = E.left$, $G.right = E.right$ }

$E \rightarrow E + T$ { $E.op = +$, $E.val = 0$, $E.left = E$,
 $E.right = T$ }

$E \rightarrow E - T$ { $E.op = -$, $E.val = 1$, $E.left = E$,
 $E.right = T$ }

$E \rightarrow T$ { $E.op = T.op$, $E.val = T.val$, $E.left = T.left$,
 $E.right = T.right$ }

$T \rightarrow T * F$ { $T.op = *$, $T.val = 2$, $T.left = T$, $T.right = F$ }

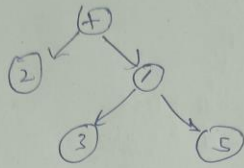
$T \rightarrow T / F$ { $T.op = /$, $T.val = 3$, $T.left = T$, $T.right = F$ }

$T \rightarrow F$ { $T.op = F.op$, $T.val = F.val$, $T.left = F.left$,
 $T.right = F.right$ }

$F \rightarrow num$ { $F.op = 0$, $F.val = num.lexval$ }

$F \rightarrow (E)$ { $F.op = E.op$, $F.val = E.val$, $F.right = E.right$,
 $F.left = E.left$ }

example:- $2 + (3/5)$



Preorder:- $+, 2, /, 3, 5$

Inorder:- $2, +, 3, /, 5$

Postorder:- $2, 3, 5, /, +$

Command to run

Yacc AST.y

Flex AST.l

Gcc y.tab.c

./a.out

2)

Command to run

Yacc DAG.y

Flex DAG.l

Gcc y.tab.c

./a.out

Input : ((a+b)*(a-b))/(c+d)-(x-y)

```
Result
node: a
node: b
node: +
node: a (Already exists)
node: b (Already exists)
node: -
node: *
node: c
node: d
node: +
node: /
node: x
node: y
node: -
node: -
```

3)

Command to run

Yacc 3ac.y

Flex 3ac.l

Gcc y.tab.c

./a.out

non terminals: G, E, T, F

terminals: $+, -, *, /, (,)$

$G \rightarrow E \quad \{ G.code = E.code \}$

$E \rightarrow E + T$

$E \rightarrow T \quad \{ E.code = T.code \}$

$E \rightarrow E_1 - T \quad \begin{cases} E.code = \\ E_1.code + E_2.code + 'E.val = \\ SUB(E.val, T.val) \end{cases}$

$E \rightarrow E_1 + T \quad \{ E.code = E_1.code + E_2.code + \\ 'E.val = ADD(E.val, T.val) \}$

$T_1 \rightarrow T_1 * F \quad \{ T.code = T_1.code + F.code + \\ 'T.val = MUL(T_1.val, F.val) \}$

$T \rightarrow T / F \quad \{ T.code = T_1.code + F.code + \\ 'T.val = MUL(T_1.val, F.val) \}$

$T \rightarrow F \quad \{ T.code = F.code \}$

$F \rightarrow (E) \quad \{ F.code = E.code \}$

$F \rightarrow num \quad \{ F.val = num.lex.val \}$

example:-

$$(a+b)*(a-b)/(c+d)-(x-y)$$

A := ADD(a,b)

B := SUB(a,b)

C := MUL(A,B)

D := ADD(c,d)

E := DIV(C,D)

F := SUB(x,y)

G := SUB(E,F)

Enter expression with +,-,*,/
((a+b)*(a-b))/(c+d)-(x-y)

The resulting 3-address code for given expression

1. A <- ADD(a,b)
2. B <- SUB(a,b)
3. C <- MUL(A,B)
4. D <- ADD(c,d)
5. E <- DIV(C,D)
6. F <- SUB(x,y)
7. G <- SUB(E,F)

given expression is valid