**OUTPUT:**

void is keyword

main is indentifier

int is keyword

abc is indentifier

= is operator

+ is operator

cab is indentifier

**OUTPUT:**

Enter a regular Expression: a|b

SIMULATION OF NFA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter a regular expression: a|b

0 1 2 3 4 5

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

E E | 0

a | 1

E | 2

b | 3

E | 4

| 5

Start state: 0

Final state: 5

**OUTPUT:**

STATES OF NFA : A, B, C, D, E, F,

GIVEN SYMBOLS FOR NFA: 0, 1, eps

NFA STATE TRANSITION TABLE

STATES |0 |1 eps

--------+------------------------------------

A |FC |- |BF

B |- |C |-

C |- |- |D

D |E |A |-

E |A |- |BF

F |- |- |-

e-Closure (A) : ABF

e-Closure (B) : B

e-Closure (C) : CD

e-Closure (D) : D

e-Closure (E) : BEF

e-Closure (F) : F

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFA TRANSITION STATE TABLE

STATES OF DFA : ABF, CDF, CD, BEF, GIVEN SYMBOLS FOR DFA: 0, 1,

STATES |0 |1

--------+-----------------------

ABF |CDF |CD

CDF |BEF |ABF

CD |BEF |ABF

BEF |ABF |CD

**OUTPUT:**

**For Left Recursion:**

Enter the productions: E->E+E|T

E+E|T The productions after eliminating Left Recursion are:

E->+EE'

E'->TE'

E->ε

**For Left Factoring:**

Enter the Parent Non-Terminal : L

Enter the number of productions : 4

Enter Production 1 : i

Enter Production 2 : iL

Enter Production 3 : (L)

Enter Production 4 : iL+L

Production Rule : L->i|iL|(L)|iL+L

After Left Factoring :

L->iL'|(L)

L'->#|LL''

L''->#|+L

**OUTPUT:**

First(E) = { (, i, }

First(R) = { +, #, }

First(T) = { (, i, }

First(Y) = { \*, #, }

First(F) = { (, i, }

-----------------------------------------------

Follow(E) = { $, ), }

Follow(R) = { $, ), }

Follow(T) = { +, $, ), }

Follow(Y) = { +, $, ), }

Follow(F) = { \*, +, $, ), }