**Exp 3:**

**Write the lex and yacc program to simulate the working of Deterministic Finite Automata (DFA)**

**Lex Code:**

%{

#include "y.tab.h"

%}

%%

a { return 'a'; }

b { return 'b'; }

. { return yytext[0]; } // Any other character is returned as-is

%%

int yywrap() {

return 1;

}

**Yacc Code:**

%{

#include <stdio.h>

#include <stdlib.h>

int yylex();

void yyerror(char \*s) {

printf("Error: %s\n", s);

}

%}

%token 'a'

%token 'b'

%%

input:

/\*space\*/ | input string

{

if ($2 == 2)

printf("Valid string (ends with 'ab')\n");

else

printf("Invalid string (does not end with 'ab')\n");

}

;

string:

{ $$ = 0; } // Starting state q0

| string 'a' { $$ = ($1 == 0 || $1 == 1) ? 1 : 0; } // Transition to q1 if 'a' from q0 or q1

| string 'b' { $$ = ($1 == 1) ? 2 : 0; } // Transition to q2 if 'b' from q1, else reset to q0

;

%%

int main() {

printf("Enter a string:\n");

yyparse();

return 0;

}

Output:



