**Program 9**

**Objective: Design a DFA in LEX Code which accepts string with even number of 0 over input alphabet {0,1}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A end

%%

<INITIAL>0 BEGIN A;

<INITIAL>1 BEGIN INITIAL;

<INITIAL>\n BEGIN INITIAL; {printf ("\nnot accepted\n");}

<A>0 BEGIN end;

<A>1 BEGIN A;

<A>\n BEGIN INITIAL; {printf ("\nnot accepted\n");}

<end>1 BEGIN end;

<end>0 BEGIN A;

<end>\n BEGIN INITIAL; {printf("\naccepted\n");}

%%

int yywrap(void)

{return 0;}

int main ()

{ yylex(); return 0; }

**OUTPUT**

**Text

Description automatically generated**

**Program 10**

**Objective: Design a DFA in LEX Code which accepts string starting with 0 and ending with 1 over input alphabet {0,1}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A end dead

%%

<INITIAL>0 BEGIN A;

<INITIAL>1 BEGIN dead;

<INITIAL>\n BEGIN INITIAL; {printf ("\nnot accepted\n");}

<A>0 BEGIN A;

<A>1 BEGIN end;

<A>\n BEGIN INITIAL; {printf ("\nnot accepted\n");}

<end>1 BEGIN end;

<end>0 BEGIN A;

<end>\n BEGIN INITIAL; {printf("\naccepted\n");}

<dead>0 BEGIN dead;

<dead>1 BEGIN INITIAL;

<dead>\n BEGIN INITIAL; {printf ("\nnot accepted\n");}

%%

int yywrap(void) {return 0;}

int main ()

{

yylex ();

return 0;

}

**OUTPUT**

**Text

Description automatically generated**

**Program 11**

**Objective:Design a DFA in LEX Code which accepts string starting with 11 over input alphabet {0,1}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A end dead

%%

<INITIAL>0 BEGIN dead;

<INITIAL>1 BEGIN A;

<INITIAL>\n BEGIN INITIAL; {printf ("not accepted\n");}

<A>0 BEGIN dead;

<A>1 BEGIN end;

<A>\n BEGIN INITIAL; {printf ("not accepted\n");}

<end>1 BEGIN end;

<end>0 BEGIN end;

<end>\n BEGIN INITIAL; {printf("accepted\n");}

<dead>0 BEGIN dead;

<dead>1 BEGIN dead;

<dead>\n BEGIN INITIAL; {printf ("not accepted\n");}

%%

int yywrap(void){return 0;}

int main ()

{

yylex ();

return 0;

}

**OUTPUT**

Text

Description automatically generated

**Program 12**

**Objective:Design a DFA in LEX Code which accepts string starting with odd 0 and even 1 over input alphabet {0,1}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A B C DEAD

%%

<INITIAL>1 BEGIN A;

<INITIAL>0 BEGIN B;

<INITIAL>[^01\n] BEGIN DEAD;

<INITIAL>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<A>1 BEGIN INITIAL;

<A>0 BEGIN C;

<A>[^01\n] BEGIN DEAD;

<A>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<B>1 BEGIN C;

<B>0 BEGIN INITIAL;

<B>[^01\n] BEGIN DEAD;

<B>\n BEGIN INITIAL; {printf("Accepted\n");}

<C>1 BEGIN B;

<C>0 BEGIN A;

<C>[^01\n] BEGIN DEAD;

<C>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<DEAD>[^\n] BEGIN DEAD;

<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}

%%

int yywrap () {return 0;}

int main ()

{

printf ("Enter String\n");

yylex ();

return 0;

}

**OUTPUT**

**Text

Description automatically generated**

**Program 13**

**Objective:Design a DFA in LEX Code which accepts string with even a and even b over input alphabet {a,b}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A B end

%%

<INITIAL>a BEGIN A;

<INITIAL>b BEGIN end;

<INITIAL>\n BEGIN INITIAL; {printf("accepted\n");}

<A>a BEGIN INITIAL;

<A>b BEGIN B;

<A>\n BEGIN INITIAL; {printf ("not accepted\n");}

<B>b BEGIN A;

<B>a BEGIN end;

<B>\n BEGIN INITIAL; {printf ("not accepted\n");}

<end>b BEGIN INITIAL;

<end>a BEGIN B;

<end>\n BEGIN INITIAL; {printf ("not accepted\n");}

%%

int yywrap(void) {return 0;}

int main ()

{

yylex ();

return 0;

}

**OUTPUT**

**Text

Description automatically generated**

**Program 14**

**Objective:Design a DFA in LEX Code which accepts string with 3rd last word from RHS be a over input alphabet {a,b}.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A B C D E F G DEAD

%%

<INITIAL>b BEGIN INITIAL;

<INITIAL>a BEGIN A;

<INITIAL>[^ab\n] BEGIN DEAD;

<INITIAL>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<A>b BEGIN F;

<A>a BEGIN B;

<A>[^ab\n] BEGIN DEAD;

<A>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<B>b BEGIN D;

<B>a BEGIN C;

<B>[^ab\n] BEGIN DEAD;

<B>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<C>b BEGIN D;

<C>a BEGIN C;

<C>[^ab\n] BEGIN DEAD;

<C>\n BEGIN INITIAL; {printf("Accepted\n");}

<D>b BEGIN G;

<D>a BEGIN E;

<D>[^ab\n] BEGIN DEAD;

<D>\n BEGIN INITIAL; {printf("Accepted\n");}

<E>b BEGIN F;

<E>a BEGIN B;

<E>[^ab\n] BEGIN DEAD;

<E>\n BEGIN INITIAL; {printf("Accepted\n");}

<F>b BEGIN G;

<F>a BEGIN E;

<F>[^ab\n] BEGIN DEAD;

<F>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<G>b BEGIN INITIAL;

<G>a BEGIN A;

<G>[^ab\n] BEGIN DEAD;

<G>\n BEGIN INITIAL; {printf("Accepted\n");}

<DEAD>[^\n] BEGIN DEAD;

<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}

%%

int yywrap()

{

return 1;

}

int main ()

{

yylex ();

return 0;

}

**OUTPUT**

Text

Description automatically generated

**Program 15**

**Objective:Design a DFA in LEX Code to Identify and print Integer & Float Constants and Identifier.**

**Code**

%{

#include<stdio.h>

#include<string.h>

%}

%s A B C DEAD

%%

<INITIAL> [0-9] + BEGIN A;

<INITIAL> [0-9] +[.] [0-9] + BEGIN B;

<INITIAL>[A-Za-z\_] [A-Za-z0-9\_] \* BEGIN C;

<INITIAL>[^\n] BEGIN DEAD;

<INITIAL>\n BEGIN INITIAL; {printf ("Not Accepted\n");}

<A>[^\n] BEGIN DEAD;

<A>\n BEGIN INITIAL; {printf("Integer\n");}

<B>[^\n] BEGIN DEAD;

<B>\n BEGIN INITIAL; {printf("Float\n");}

<C>[^\n] BEGIN DEAD;

<C>\n BEGIN INITIAL; {printf("Identifier\n");}

<DEAD>[^\n] BEGIN DEAD;

<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}

%%

int yywrap () { return 1; }

int main ()

{

yylex ();

return 0;

}

**OUTPUT**

Text

Description automatically generated