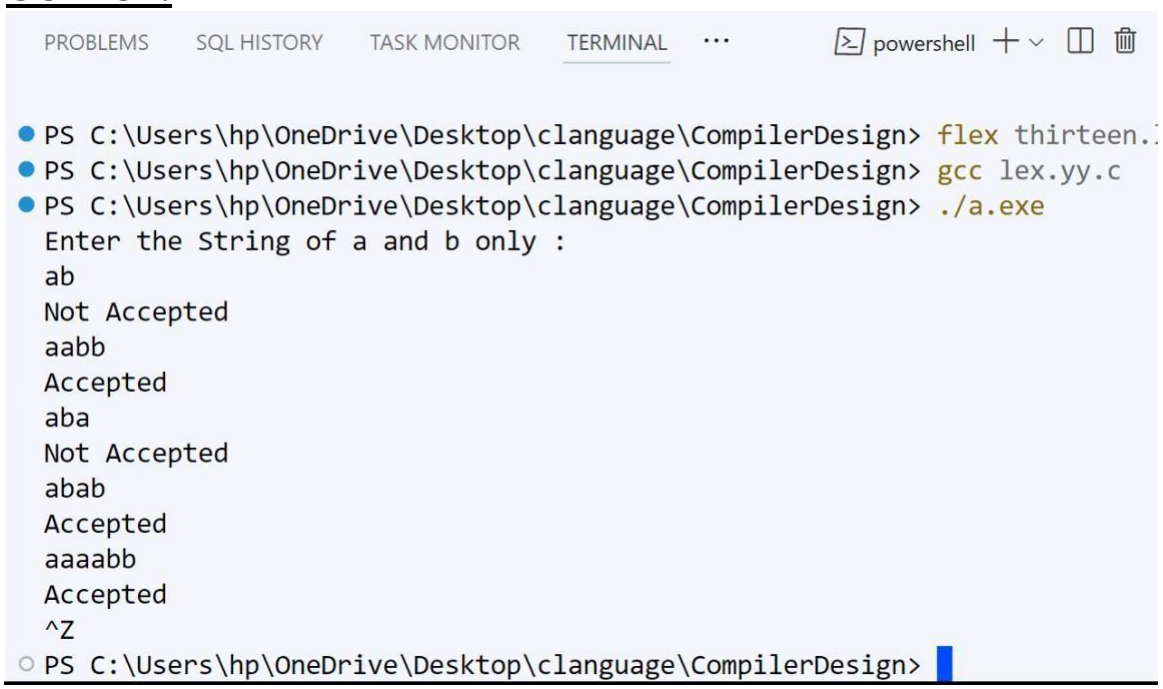
13: Design a LEX Code which accepts all possible set of string containing even number of ‘a’ and even number of ‘b’ over input alphabet ∑={a,b}.

Source Code:

```
%{
#include<stdio.h>
%}
%s A B C F
%%
<INITIAL>\n printf("\nAccepted\n\n");BEGIN INITIAL;
<INITIAL>a BEGIN A;
<INITIAL>b BEGIN B;
<A>a BEGIN INITIAL;
<A>b BEGIN C;
<A>\n BEGIN INITIAL; printf("\nNot Accepted\n");
<B>a BEGIN C;
<B>b BEGIN INITIAL;
<B>\n BEGIN INITIAL; printf("\nNot Accepted\n");
<C>a BEGIN B;
<C>b BEGIN A;
<C>\n BEGIN INITIAL; printf("\nNot Accepted\n");
<A>[^ab\n] BEGIN F;
<B>[^ab\n] BEGIN F;
<C>[^ab\n] BEGIN F;
<INITIAL>[^ab\n] BEGIN F;
<F>[^\n] BEGIN F;
<F>[\n] BEGIN INITIAL;printf("\nInvalid Input\n");
.;
%%

int yywrap()
{ return 1;
}
int main()
{ printf("Enter the String of a and b only :
   \n"); yylex();
}
```

OUTPUT:


PROBLEMS
SQL HISTORY
TASK MONITOR
TERMINAL
powershell
PS C:\Users\hp\OneDrive\Desktop\clanguage\CompilerDesign> flex thirteen.
PS C:\Users\hp\OneDrive\Desktop\clanguage\CompilerDesign> gcc lex.yy.c
PS C:\Users\hp\OneDrive\Desktop\clanguage\CompilerDesign> ./a.exe
Enter the String of a and b only :
ab
Not Accepted
aabb
Accepted
aba
Not Accepted
abab
Accepted
aaaabb
Accepted
^Z
PS C:\Users\hp\OneDrive\Desktop\clanguage\CompilerDesign>

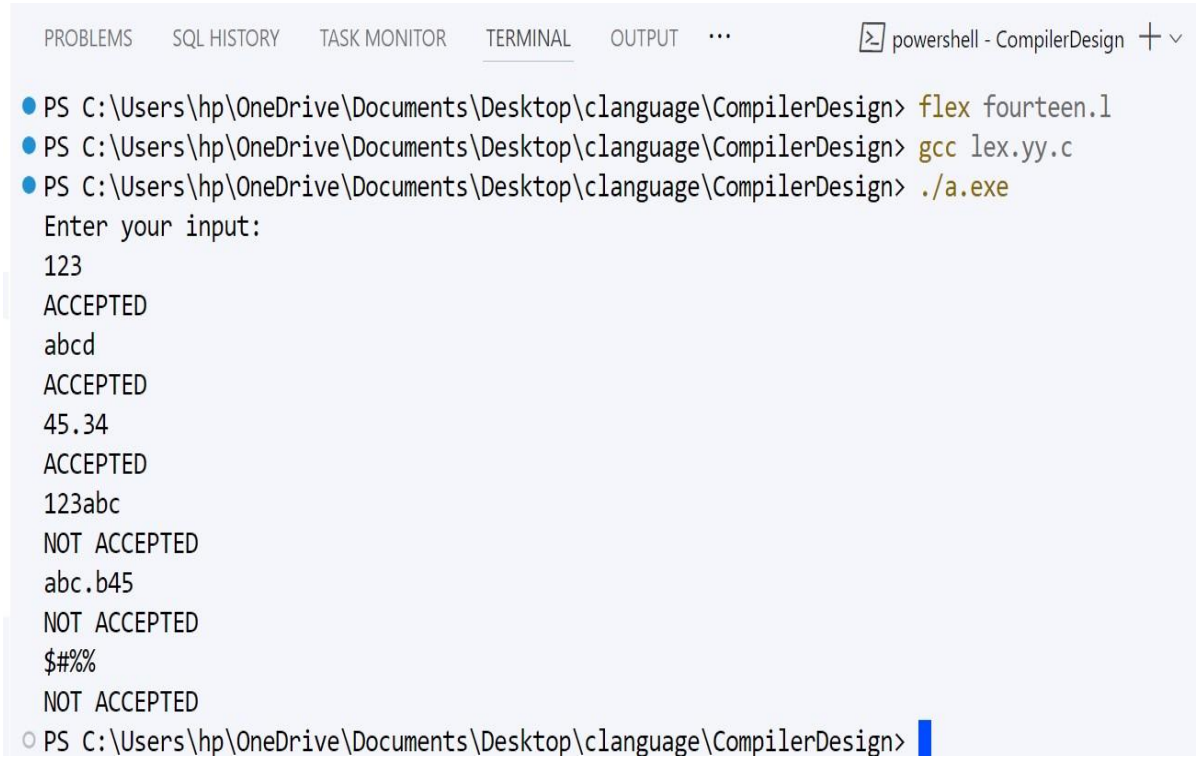
14: Design a DFA in LEX Code to Identify and print Integer, Float Constants and Identifier.

Source Code:

```
%{
  #include<stdio.h>
%}
%s A B C D E
%%
<INITIAL>\n   { printf("NOT ACCEPTED\n"); BEGIN(INITIAL); }
<INITIAL>-?[0-9]+ { BEGIN(A); }
<INITIAL>[a-zA-Z_][a-zA-Z0-9]* { BEGIN(D); }
<INITIAL>[^\n]    { BEGIN(E); }
<A>\n           { printf("ACCEPTED\n"); BEGIN(INITIAL); }
<A>[.]          { BEGIN(B); }
<A>[^\n]          { BEGIN(E); }
<B>[0-9]+         { BEGIN(C); }
<B>[^0-9\n]       { BEGIN(E); }
<C>\n           { printf("ACCEPTED\n"); BEGIN(INITIAL); }
<C>[^\n]          { BEGIN(E); }
<D>\n           { printf("ACCEPTED\n"); BEGIN(INITIAL); }
<D>[^\n]          { BEGIN(E); }
<E>\n           { printf("NOT ACCEPTED\n"); BEGIN(INITIAL); }
<E>[^\n]         { BEGIN(E); }
%%
int main()
{
   printf("Enter your
input:\n");     yylex();
return 0;
}

int
yywrap() {
return 1;
}
```

OUTPUT:

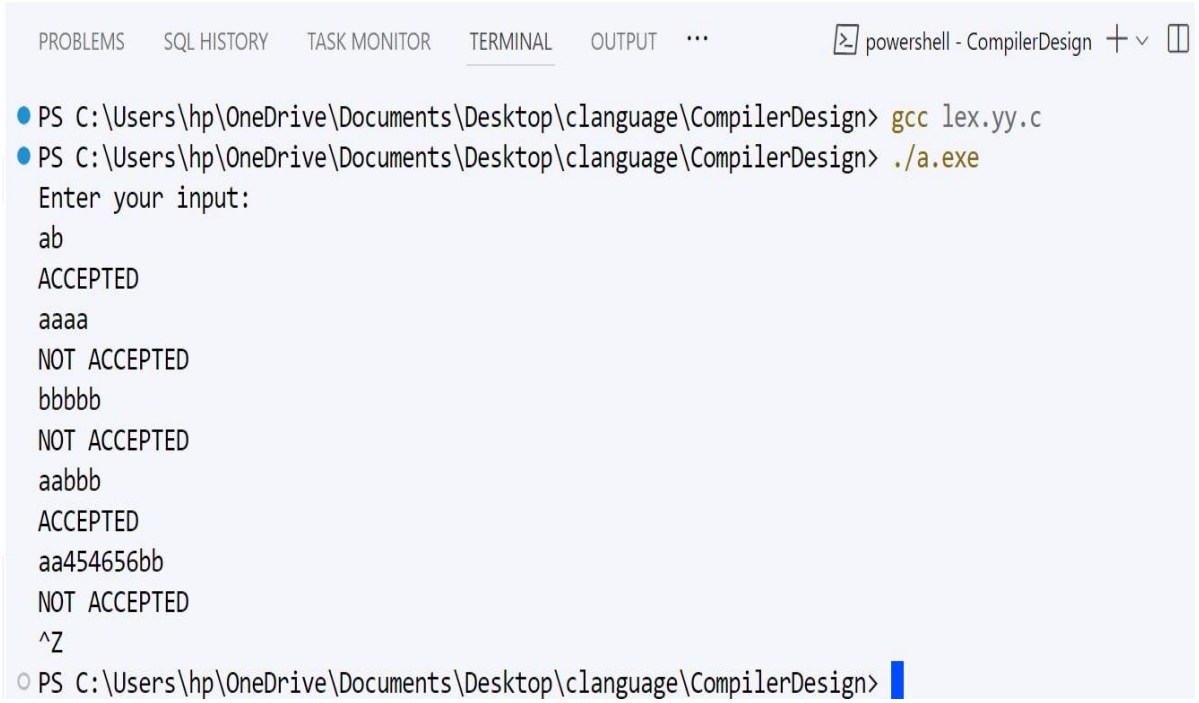

PROBLEMS
SQL HISTORY
TASK MONITOR
TERMINAL
OUTPUT
powershell - CompilerDesign
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign> flex fourteen.l
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign> gcc lex.yy.c
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign> ./a.exe
Enter your input:
123
ACCEPTED
abcd
ACCEPTED
45.34
ACCEPTED
123abc
NOT ACCEPTED
abc.b45
NOT ACCEPTED
$#%%
NOT ACCEPTED
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign>

15: Design a DFA in LEX Code over Σ = {a, b} which contains set of all possible strings where every string starts with a and ends with b.

Source Code:

```
%{
#include<stdio.h>
%}
%s A B C
%%
<INITIAL>[a]+ {BEGIN A;}
<INITIAL>[\n] {printf("NOT ACCEPTED\n"); BEGIN INITIAL;}
<INITIAL>[^a] {BEGIN C;}
<A>[b]+ {BEGIN B;}
<A>[\n] {printf("NOT ACCEPTED\n"); BEGIN INITIAL;}
<A>[^ab] {BEGIN C;}
<B>[\n] {printf("ACCEPTED\n");BEGIN INITIAL;}
<B>[a]+ {BEGIN A;}
<B>[^ab\n] {BEGIN C;}
<C>[^\n] {BEGIN C;}
<C>[\n] {printf("NOT ACCEPTED\n");BEGIN INITIAL;}
%%
int main()
{
   printf("Enter your
input:\n");     yylex();
return 0;
} int
yywrap()
{return 1;}
```

OUTPUT:


PROBLEMS
SQL HISTORY
TASK MONITOR
TERMINAL
OUTPUT
powershell - CompilerDesign
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign> gcc lex.yy.c
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign> ./a.exe
Enter your input:
ab
ACCEPTED
aaaa
NOT ACCEPTED
bbbbb
NOT ACCEPTED
aabbb
ACCEPTED
aa454656bb
NOT ACCEPTED
^Z
PS C:\Users\hp\OneDrive\Documents\Desktop\clanguage\CompilerDesign>