**Practical-3**

1. **Write a Lex program to recognize identifiers in C**
   * **C identifiers should have following constraint**
     + **It should start with either letter or underscore (\_) sign.**
     + **It should not contain special symbols.**

%{

#include<stdio.h>

%}

%%

^[a-zA-Z\_][\_a-zA-Z0-9]+ {printf("Identifier\n");}

.\* {printf("Not a Identifier\n");}

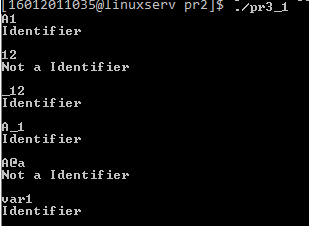
%%

main()

{

yylex();

}



1. **Write a Lex program for validation of Email-Add. (Consider Email Add. from any domain e.g. @gmail.com)**

%{

#include<stdio.h>

%}

%%

^[a-z][a-z0-9\_]\*(@[A-Za-z]+)(\.[a-z]+)+ {printf("valid");}

.\* {printf("invalid");}

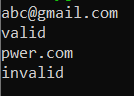
%%

main()

{

yylex();

}



1. **Write a Lex program to identify integer, float and exponential value.**

Examples:

123

Integer

12.23

Float

12E23

Exponential

12.25E23

Exponential

12.25E-25

Exponential

-12.25E-25

Exponential

12.25E25.25

Other (Reason: float value after E not allowed)

%{

#include<stdio.h>

%}

%%

[0-9]+printf("Integer");

[0-9]+\.[0-9]+ printf("Float");

[0-9]+E[?\-0-9][0-9]\* printf("Exponential");

[?\-0-9][0-9]\*?.[0-9]+E[?\-0-9][0-9]\* printf("Exponential");

.\*printf("Other");

%%

int yywrap(void)

{

return 1;

}

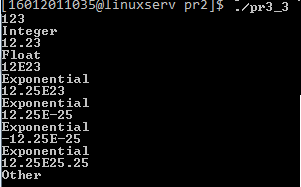
int main()

{

yylex();

return 0;

}



1. **Implement a Lex program that takes input from user specified file and labels each input as Integer, Float, Exponential or Other.**

Consider file test.txt contains following data:

123

123.25

15

Abs

12.25E25

Expected Output:

123 - Integer

123.25 - Float

15 - Integer

Abs - Other

12.25E25 - Exponential

#include<stdio.h>

%}

%%

/\*\*\* Rules section \*\*\*/

[-0-9]\* {

printf("Integer Value");

}

^[-|0-9][0-9]+(.)[0-9]+ {

printf("Float Value");

}

[-0-9]\*(.)[0-9]+(E)[-0-9]+ {

printf("Exponential Value");

}

.\* {

printf("Invalid value");

}

%%

int main()

{

FILE \*fp;

fp = fopen("test1.txt", "r");

if (fp == NULL) { printf("File not found"); }

yyin = fp;

yylex();

return 0;

}

