/* A4: Write a program in LEX to recognize different tokes: Keywords, Identifiers, Constants, Operators and Punctuations */

```
d[0-9]
a[A-Za-z]
z[a-zA-Z0-9]
x[.]

%{
      /*
        d recognizes a digit
        a recognizes alphabet both capital and small
        z recognizes alphabets and digits
        x recognizes dot
      */
        int x1,x2,x3,x4,x5;
      /* x1 is counter for keywords, x2 for numbers, x3 for
identifiers, x4 for operators x5 for punctuations */
%}

%%

int|float|char { /* int, float, char are keywords */ x1++; }

[+-]?{d}{d}*({x}{d}{d}*)?(e[+-]?{d}+)? { /* pattern for numbers
includeing scientific representation */
                             x2++;
                             }

{a}{z}* { /* identifer always starts with an alphabet and then it
can have either alphabet or digit */
       x3++;
      }

=|>=|==|<= { /* operators */
           x4++;
         }

;|, { /* semi-colon and comma are punctuations */
      x5++;
    }

({z}|[+-]|[.]|e)* { ; }

\n {/* After every line print */
    printf("\n\nNumber of Keywords:%d\n",x1);
    printf("Number of Numbers:%d\n",x2);
    printf("Number of Identifiers:%d\n",x3);
    printf("Number of Operators:%d\n",x4);
    printf("Number of Puntuations:%d\n",x5);
```

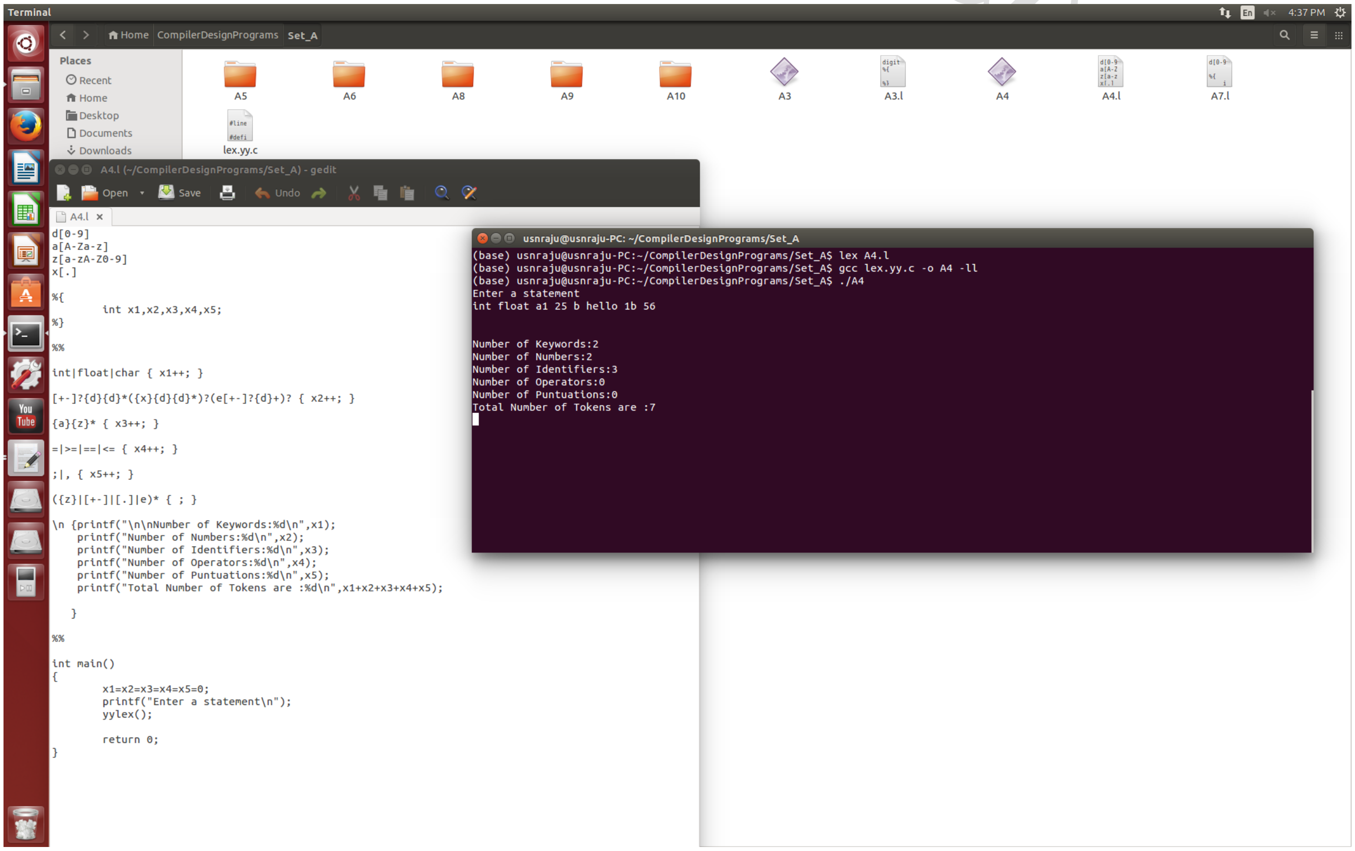
```
     printf("Total Number of Tokens are :%d\n",x1+x2+x3+x4+x5);

    }

%%

int main()
{
        x1=x2=x3=x4=x5=0;
        printf("Enter a statement\n");
        yylex();

        return 0;
}
```



```
d[0-9]
a[A-Za-z]
z[a-zA-Z0-9]
x[.]

%{
        int x1,x2,x3,x4,x5;
%}

%%

int|float|char { x1++; }

[+-]?{d}{d}*({x}{d}{d}*)?(e[+-]?{d}+)? { x2++; }

{a}{z}* { x3++; }

=|>=|==|<= { x4++; }

;|, { x5++; }

({z}|[+-]|[.]|e)* { ; }

\n {printf("\n\nNumber of Keywords:%d\n",x1);
    printf("Number of Numbers:%d\n",x2);
    printf("Number of Identifiers:%d\n",x3);
    printf("Number of Operators:%d\n",x4);
    printf("Number of Puntuations:%d\n",x5);
    printf("Total Number of Tokens are :%d\n",x1+x2+x3+x4+x5);

   }

%%

int main()
{
        x1=x2=x3=x4=x5=0;
        printf("Enter a statement\n");
        yylex();

        return 0;
}
```

```
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ lex A4.l
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ gcc lex.yy.c -o A4 -ll
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ ./A4
Enter a statement
int float a1 25 b hello 1b 56


Number of Keywords:2
Number of Numbers:2
Number of Identifiers:3
Number of Operators:0
Number of Puntuations:0
Total Number of Tokens are :7
```