/* A7: Write a LEX program to recognize the following tokens over the alphabets {0,1,..,9}

a) The set of all string ending in 00.
b) The set of all strings with three consecutive 222's.
c) The set of all string such that every block of five consecutive symbols contains at least two 5's.
d) The set of all strings beginning with a 1 which, interpreted as the binary representation of an integer, is congruent to zero modulo 5.
e) The set of all strings such that the 10th symbol from the right end is 1.
f) The set of all four digits numbers whose sum is 9
g) The set of all four digital numbers, whose individual digits are in ascending order from left to right. */

```
d[0-9]

%{
    /* d is for recognising digits */
    int c1=0,c2=0,c3=0,c4=0,c5=0,c6=0,c7=0;
    /* c1 to c7 are counters for rules a1 to a7 */
%}

%%

({d})*00 { /* Strings ending with 00 */
          c1++; printf("%s rule A\n",yytext);
       }

({d})*222({d})* { /* strings having 3 consecutive 2's which can
be written as 0 or more digits followed by 3 2's and then 0 or
more digits */
                    c2++; printf("%s rule B \n",yytext);
            }

(1(0)*(11|01)(01*01|00*10(0)*(11|1))*0)(1|10(0)*(11|01)(01*01|00*
10(0)*(11|1))*10)*  { /* Binary strings congruent to 0 (mod 5) */
                                                                c4++;
printf("%s rule D \n",yytext);
                                                               }

({d})*1{d}{9} { /* All strings with 1 as the 10th digit from the
end. The strings can begin with anythiong followed by a 1
followed by exactly 9 characters */
            c5++; printf("%s rule E \n",yytext);
             }

{d}{4} { /* all 4 digit numbers */
                      int sum=0,i; for(i=0;i<4;i++) {
sum=sum+yytext[i]-48; }
                      if(sum==9) { c6++; printf("%s rule F \n",yytext);
/* if sum is 9 */}
```

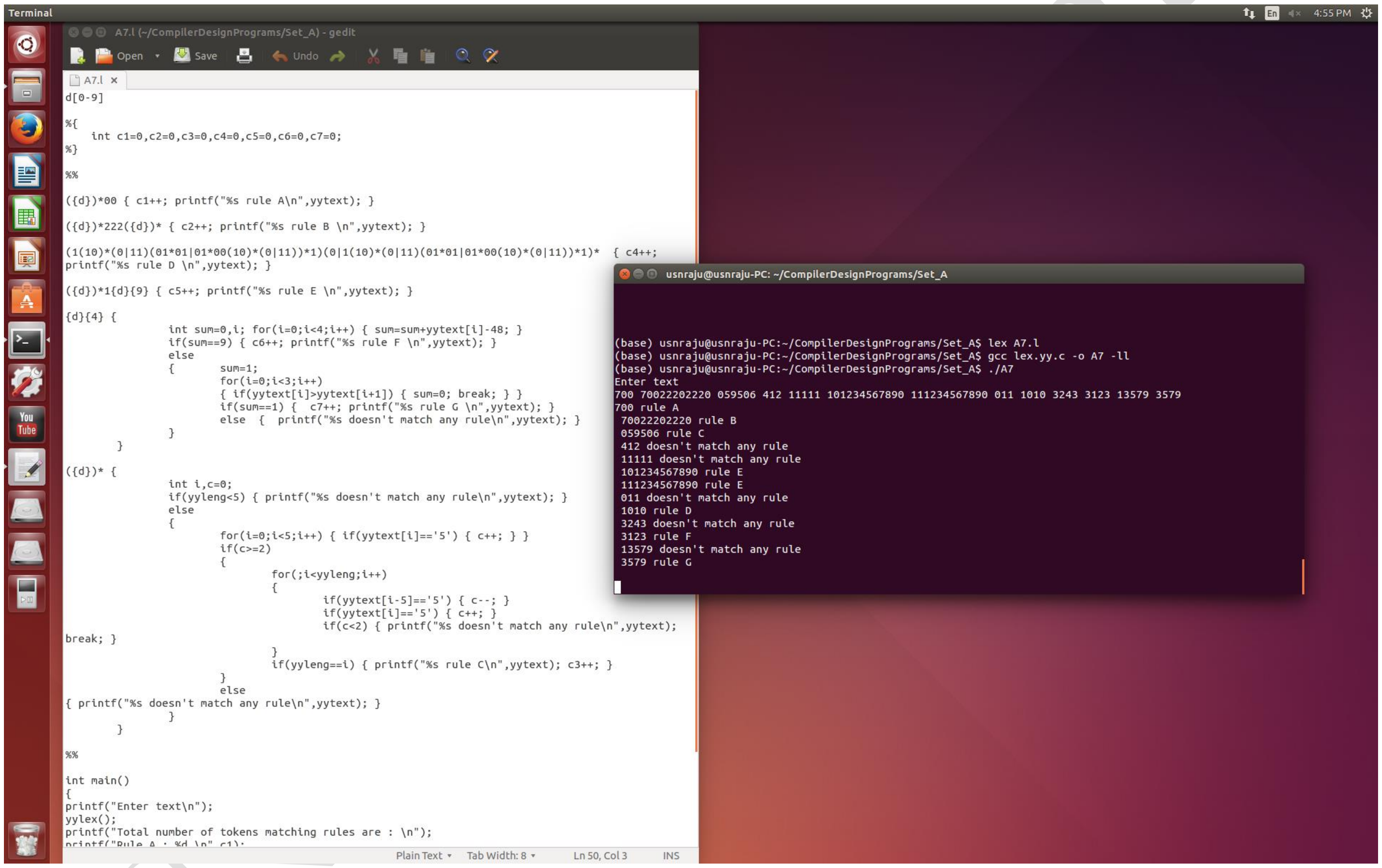
```
                else
                {       /* else check if the numbers are in
increasing order from left to right*/
                sum=1;
                        for(i=0;i<3;i++)
                        { if(yytext[i]>yytext[i+1]) { sum=0;
break; } }
                        if(sum==1) {  c7++; printf("%s rule G
\n",yytext); }
                        else  {  printf("%s doesn't match any
rule\n",yytext); }
                }
        }

({d})* {   /* all string such that every block of five
consecutive symbols contains at least two 5's */
                int i,c=0;
                if(yyleng<5) { printf("%s doesn't match any
rule\n",yytext); }
                else
                {     /* for every block find the number of 5's
*/
                        for(i=0;i<5;i++) { if(yytext[i]=='5') {
c++; } }
                        if(c>=2)
                        {
                                for(;i<yyleng;i++)
                                {
                                        if(yytext[i-5]=='5') { c-
-; } /* A block is complete so decrease counter */
                                        if(yytext[i]=='5') { c++;
}
                                        if(c<2) { printf("%s
doesn't match any rule\n",yytext);  break; }
                                }
                                if(yyleng==i) { printf("%s rule
C\n",yytext); c3++; }
                        }
                        else
                {
                      printf("%s doesn't match any
rule\n",yytext);
                }
                }
        }

%%

int main()
{
printf("Enter text\n");
yylex();
```

```
printf("Total number of tokens matching rules are : \n");
printf("Rule A : %d \n",c1);
printf("Rule B : %d \n",c2);
printf("Rule C : %d \n",c3);
printf("Rule D : %d \n",c4);
printf("Rule E : %d \n",c5);
printf("Rule F : %d \n",c6);
printf("Rule G : %d \n",c7);
return 0;
}
```


Terminal
A7.l (~/CompilerDesignPrograms/Set_A) - gedit
Open
Save
Undo
A7.l
Plain Text
Tab Width: 8
Ln 50, Col 3
INS
4:55 PM
usnraju@usnraju-PC: ~/CompilerDesignPrograms/Set_A

```
d[0-9]

%{
   int c1=0,c2=0,c3=0,c4=0,c5=0,c6=0,c7=0;
%}

%%

({d})*00 { c1++; printf("%s rule A\n",yytext); }

({d})*222({d})* { c2++; printf("%s rule B \n",yytext); }

(1(10)*(0|11)(01*01|01*00(10)*(0|11))*1)(0|1(10)*(0|11)(01*01|01*00(10)*(0|11))*1)*  { c4++;
printf("%s rule D \n",yytext); }

({d})*1{d}{9} { c5++; printf("%s rule E \n",yytext); }

{d}{4} {
		int sum=0,i; for(i=0;i<4;i++) { sum=sum+yytext[i]-48; }
		if(sum==9) { c6++; printf("%s rule F \n",yytext); }
		else
		{	sum=1;
			for(i=0;i<3;i++)
			{ if(yytext[i]>yytext[i+1]) { sum=0; break; } }
			if(sum==1) {  c7++; printf("%s rule G \n",yytext); }
			else  {  printf("%s doesn't match any rule\n",yytext); }
		}
	}

({d})* {
		int i,c=0;
		if(yyleng<5) { printf("%s doesn't match any rule\n",yytext); }
		else
		{
			for(i=0;i<5;i++) { if(yytext[i]=='5') { c++; } }
			if(c>=2)
			{
				for(;i<yyleng;i++)
				{
					if(yytext[i-5]=='5') { c--; }
					if(yytext[i]=='5') { c++; }
					if(c<2) { printf("%s doesn't match any rule\n",yytext);
break; }
				}
				if(yyleng==i) { printf("%s rule C\n",yytext); c3++; }
			}
			else
{ printf("%s doesn't match any rule\n",yytext); }
		}
	}

%%

int main()
{
printf("Enter text\n");
yylex();
printf("Total number of tokens matching rules are : \n");
```

```
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ lex A7.l
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ gcc lex.yy.c -o A7 -ll
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_A$ ./A7
Enter text
700 70022202220 059506 412 11111 101234567890 111234567890 011 1010 3243 3123 13579 3579
700 rule A
 70022202220 rule B
 059506 rule C
 412 doesn't match any rule
 11111 doesn't match any rule
 101234567890 rule E
 111234567890 rule E
 011 doesn't match any rule
 1010 rule D
 3243 doesn't match any rule
 3123 rule F
 13579 doesn't match any rule
 3579 rule G
```