# Department of Computer Science and Engineering,SVNIT Surat
## Lab Assignment 4

**U20CS005**
**BANSI MARAKANA**

**1. Write a Lex program to count the number of lines, characters and words of the given input file.**

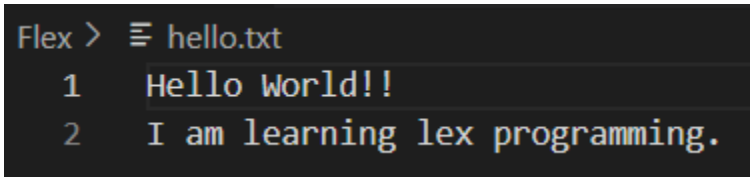
```
%{
int line_count = 1, word_count = 0, char_count = 0;
%}

%%
\n {line_count++; char_count++;}
[^ \t\n]+ {word_count++; char_count+=yyleng;}
. {char_count++;}
%%

int main()
{
   char fname[20];
   printf("Enter file name: ");
   scanf("%s",fname);
   yyin = fopen(fname,"r");
         yylex();
         printf("Number of lines: %d\n",line_count);
         printf("Number of words: %d\n",word_count);
   printf("Number of characters: %d\n",char_count);
         return 0;
}

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

Content of file hello.txt



```
PS D:\C Programs (VS Code)\Flex> flex a4q1.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter file name: hello.txt
Number of lines: 2
Number of words: 7
Number of characters: 44
```

**2. Write a lex program to find out the total number of vowels, and consonants from the given input string.**

```
%{
int vowels=0, consonants=0;
%}

%%
[aeiouAEIOU] {vowels++;}
[bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ] {consonants++;}
\n  {return 0;}
%%

int main()
{
   printf("Enter a sentence/string to count vowels and consonants:\n");
   yylex();
   printf("\nNumber of vowels: %d", vowels);
   printf("\nNumber of consonants: %d\n", consonants);
   return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q2.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter a sentence/string to count vowels and consonants:
Bansi Marakana

Number of vowels: 6
Number of consonants: 7
```

**3. Write a Lex Program to convert Lowercase string to Upper case.**
**Input: abc Output: ABC**

```
%%
[a-z] printf("%c",yytext[0] - ('a' - 'A'));
\n  {return 0;}
%%

int main()
{
   printf("Enter a sentence/string to convert it to upper case:\n");
   yylex();
   return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q3.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter a sentence/string to convert it to upper case:
Bansi Marakana
BANSI MARAKANA
```

**4. Program to count no of: a) +ve and –ve integers b) +ve and –ve fractions**

```
%{
int positivenumber=0, negativenumber=0, positivefraction=0, negativefraction=0;
%}

%%
[+]?[0-9]+ {positivenumber++;}
[-][0-9]+ {negativenumber++;}
[+]?[0-9]*\.[0-9]+ {positivefraction++;}
[-][0-9]*\.[0-9]+ {negativefraction++;}
\n  {return 0;}
%%

int main()
{
   printf("Enter numbers: ");
         yylex();
         printf("\nNumber of positive numbers: %d", positivenumber);
```

```
        printf("\nNumber of Negative numbers: %d", negativenumber);
        printf("\nNumber of Positive numbers in fractions: %d", positivefraction);
        printf("\nNumber of Negative numbers in fractions: %d\n", negativefraction);
        return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q4.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter numbers: 34 -3421 5.87 -5.989 4.0 0 2321 -34 -42.5 234.34 452 -324

Number of positive numbers: 4
Number of Negative numbers: 3
Number of Positive numbers in fractions: 3
Number of Negative numbers in fractions: 2
```

**5. Write a Lex program to check valid/invalid**
**(a) Mobile number (considering 10-digit mobile number followed by country code +91)**

```
%%
"+91"[1-9][0-9]{9} {printf("Valid Mobile Number!!");}
.+ {printf("Invalid Mobile Number!!");}
\n {return 0;}
%%

int main()
{
        printf("Enter Mobile Number: ");
        yylex();
        return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q5a.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter Mobile Number: +917845123639
Valid Mobile Number!!
PS D:\C Programs (VS Code)\Flex> ./a
Enter Mobile Number: 69333245789
Invalid Mobile Number!!
PS D:\C Programs (VS Code)\Flex> ./a
Enter Mobile Number: +910232151417
Invalid Mobile Number!!
```

**(b) Email address**

```
%%
[a-z0-9]+[a-z.0-9]+@[a-z]+(".com"|".in"|".org") {printf("Valid Email Address!!");}
.+ {printf("Invalid Email Address!!");}
\n  {return 0;}
%%

int main()
{
   printf("Enter Email Address: ");
   yylex();
         return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q5b.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter Email Address: bansimarakana@gmail.com
Valid Email Address!!
PS D:\C Programs (VS Code)\Flex> ./a
Enter Email Address: .bansi@gmail.com
Invalid Email Address!!
```

**6. Write a Lex program to implement a simple Calculator.**

```
%{
int op = 0,i;
float a, b, t;
```

```
%}

%%
[0-9]+|([0-9]*)"."([0-9]+) {cal();}
"+" {op=1;}
"-" {op=2;}
"*" {op=3;}
"/" {op=4;}
"^" {op=5;}
\n {printf("The Answer is %f\n",a); return 0;}
%%

int cal()
{
   if(op==0)
      a=atof(yytext);
   else
   {
      b=atof(yytext);
      switch(op)
      {
         case 1:
            a=a+b;
            break;
         case 2:
            a=a-b;
            break;
         case 3:
            a=a*b;
            break;
         case 4:
            a=a/b;
            break;
         case 5:
            for(i=a;b>1;b--)
               a=a*i;
            break;
      }
      op=0;
   }
   return 0;
}

int main()
```

```
{
   printf("Enter expression: ");
   yylex();
        return 0;
}

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

```
PS D:\C Programs (VS Code)\Flex> ./a
Enter expression: 3+4
The Answer is 7.000000
PS D:\C Programs (VS Code)\Flex> ./a
Enter expression: 3-5
The Answer is -2.000000
PS D:\C Programs (VS Code)\Flex> ./a
Enter expression: 4/2
The Answer is 2.000000
PS D:\C Programs (VS Code)\Flex> ./a
Enter expression: 3*568
The Answer is 1704.000000
PS D:\C Programs (VS Code)\Flex> ./a
Enter expression: 3^4
The Answer is 81.000000
```

**7. Program to recognize whether a given sentence is simple or compound.**

```
%{
        #include<stdio.h>
        int flag=0;
%}

%%
and { flag=1; }
or { flag=1; }
but { flag=1; }
because { flag=1; }
if { flag=1; }
then { flag=1; }
nevertheless { flag=1; }
.  ;
\n  { return 0; }
%%
```

```
int main()
{
	printf("Enter the sentence:\n");
	yylex();
	if(flag==0)
		printf("It is a simple sentence\n");
	else
		printf("It is a compound sentence\n");
}

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

```
PS D:\C Programs (VS Code)\Flex> flex a4q7.l
PS D:\C Programs (VS Code)\Flex> gcc lex.yy.c
PS D:\C Programs (VS Code)\Flex> ./a
Enter the sentence:
Cats hates water
It is a simple sentence
PS D:\C Programs (VS Code)\Flex> ./a
Enter the sentence:
She goes to beach and takes her water
It is a compound sentence
```