Compiler Design Lab Work

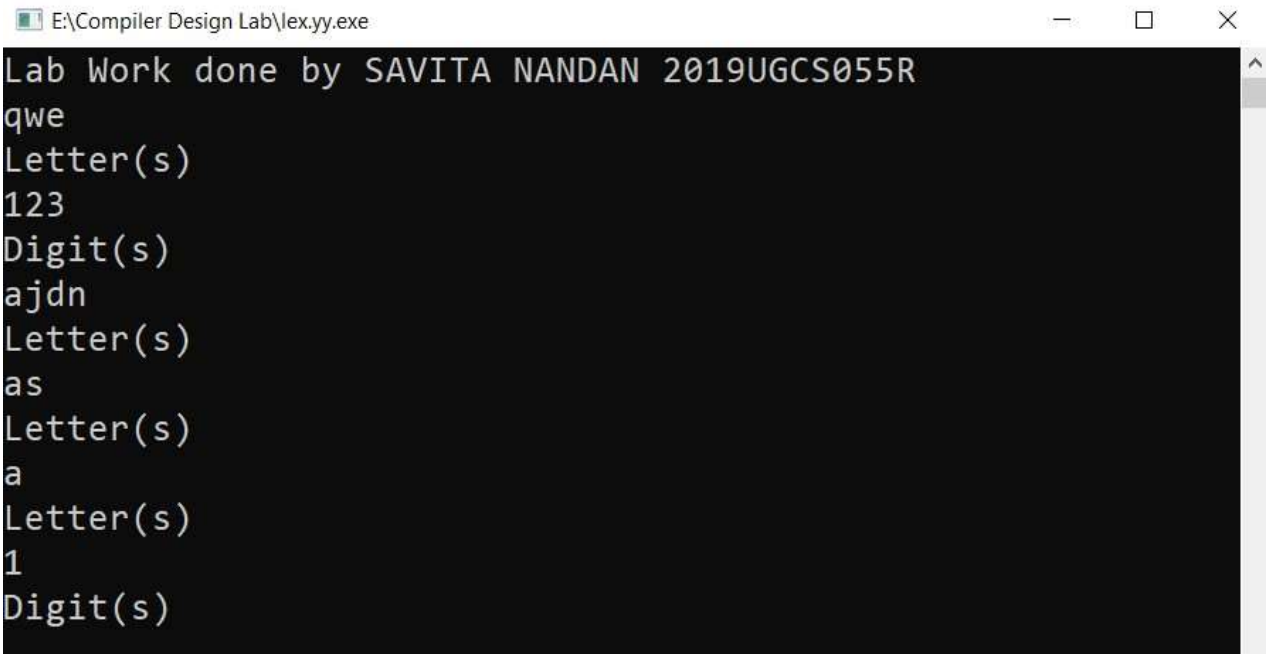
Date – 06.08.2021

Q1. Write a Lex program to distinguish between letter and digit

Code –

```
%{
#define A 100
%}
WS [ \t]+
letter [A-Za-z]
digit [0-9]
op_plus "+"

%%
{digit}+ {printf("Digit(s)");};
{letter}+ {printf("Letter(s)");};
%%

int main() {
   printf("Lab Work done by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output –


E:\Compiler Design Lab\lex.yy.exe
Lab Work done by SAVITA NANDAN 2019UGCS055R
qwe
Letter(s)
123
Digit(s)
ajdn
Letter(s)
as
Letter(s)
a
Letter(s)
1
Digit(s)

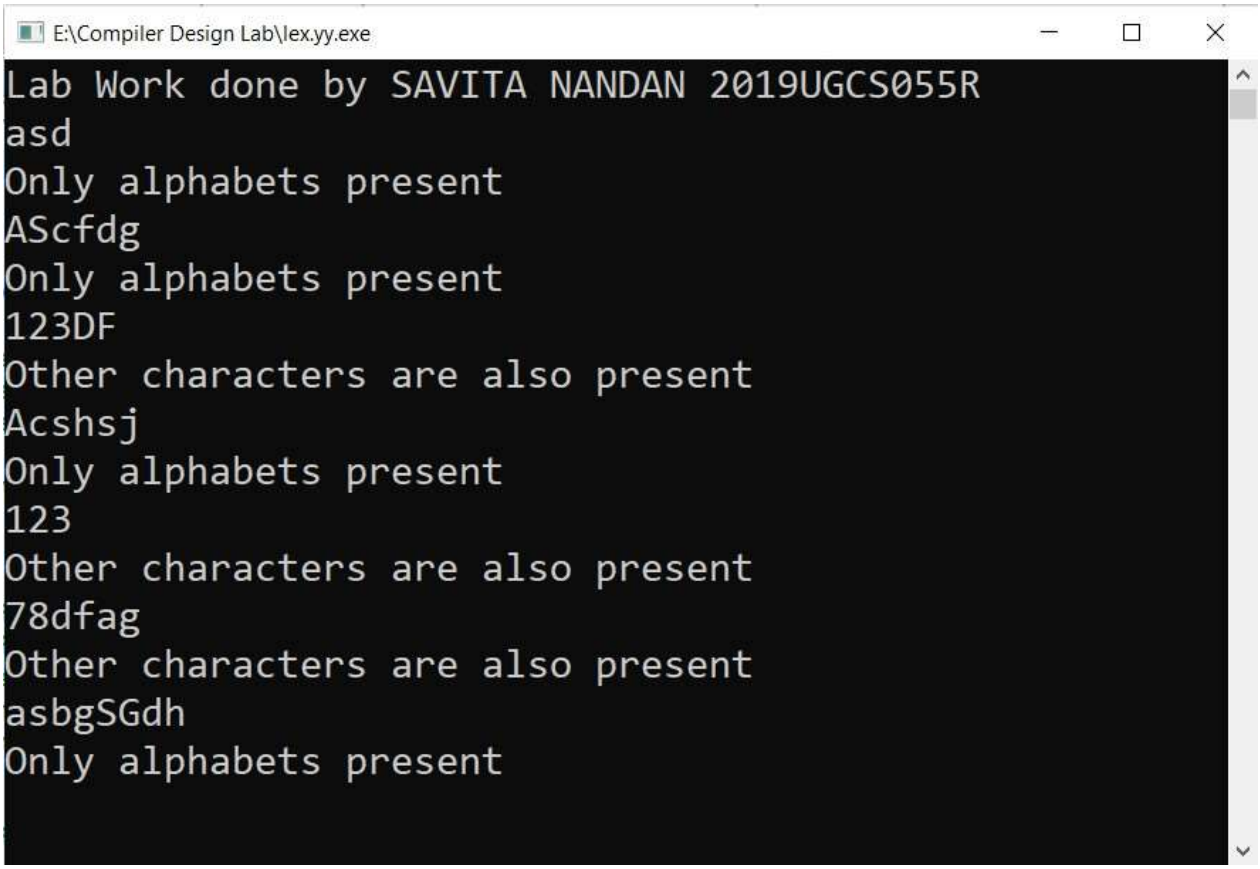
Date – 14.08.2021

Q1. Write a Lex program to check if input contain only alphabet

Code –

```
%{
 int flag = 0;
%}
%%
[\n] {
   (flag==0)?printf("Only alphabets present\n"):
   printf("Other characters are also present\n");
   flag = 0;
}

[^a-zA-Z] {flag = 1;}
. {}
%%
int main() {
   printf("Lab Work done by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output –



Q2. Write a Lex program to check if input contain only letters or digits or combination of both.

Code –

```
%{
#include<stdio.h>
#include<stdlib.h>
%}
%%
[0-9]+ {printf("Digit only");};
[A-Za-z]+ {printf("Letter only");};
^[A-Za-z]+|[0-9]+|[A-Za-z]+[0-9]*|[0-9]+[A-Za-z]* {printf("Invalid Format");};
%%
```

```
int main() {
   printf("Lab Work done by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output –

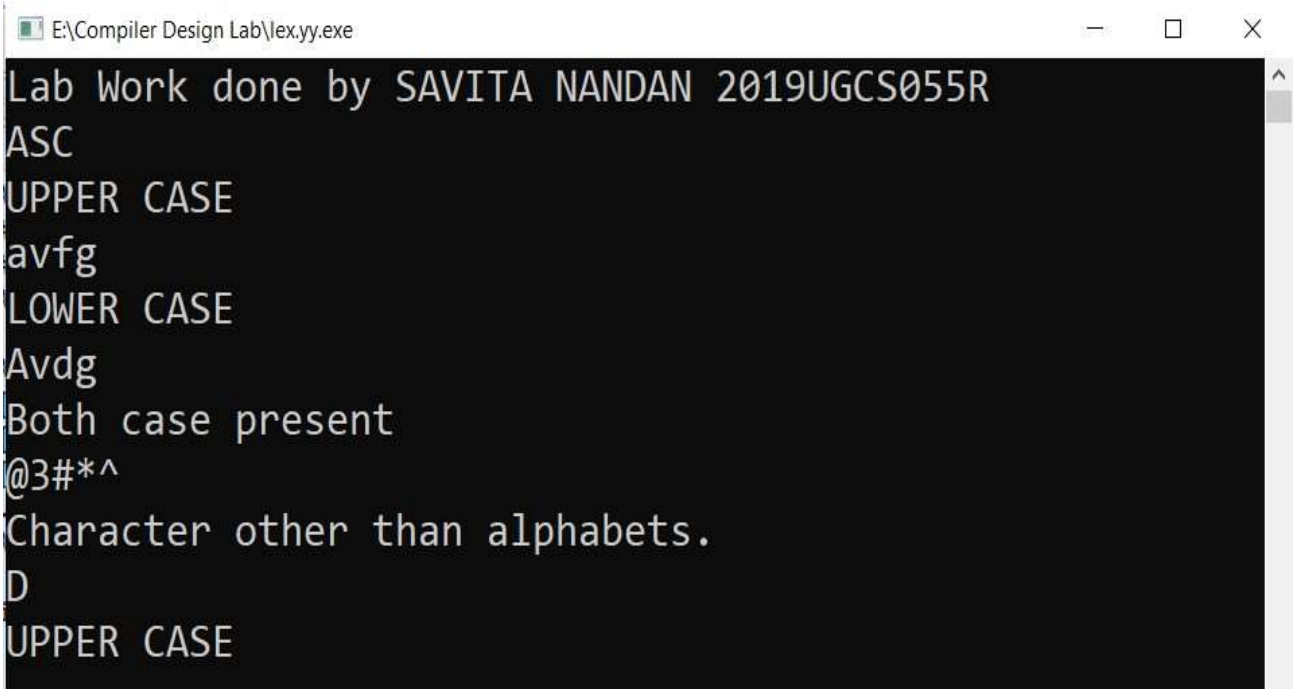
Q3. Write a Lex program to check whether the input is lower case or uppercase.

Code –

```
%{
#include<stdio.h>
```

```
#include<stdlib.h>
%}

%%
[a-z]+  printf("LOWER CASE");
[A-Z]+  printf("UPPER CASE");
[A-Za-z]+ printf("Both case present");
.* {printf("Character other than alphabets.");}
%%

int main() {
   printf("Lab Work done by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output -



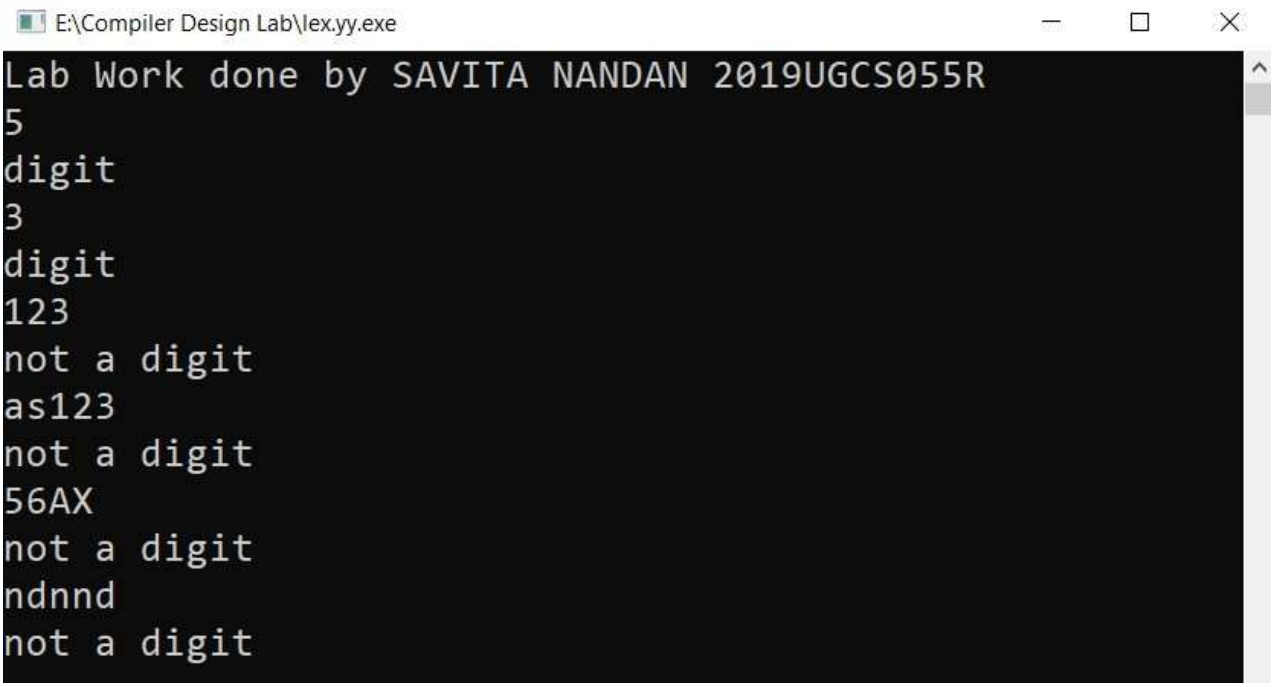
Q4. Write a Lex program to check whether the input is digit or not

Code –

```
%{
#include<stdio.h>
#include<stdlib.h>
%}

WS [ \t]+
letter [A-Za-z]
digit [0-9]
op_plus "+"
```

```
%%
[0-9]  printf("digit");
^[^0-9]|[0-9]*[a-zA-Z]*|[0-9]*|[a-zA-Z]*[0-9]*  printf("not a digit");
. ;
%%

int main() {
   printf("Lab Work done by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output –



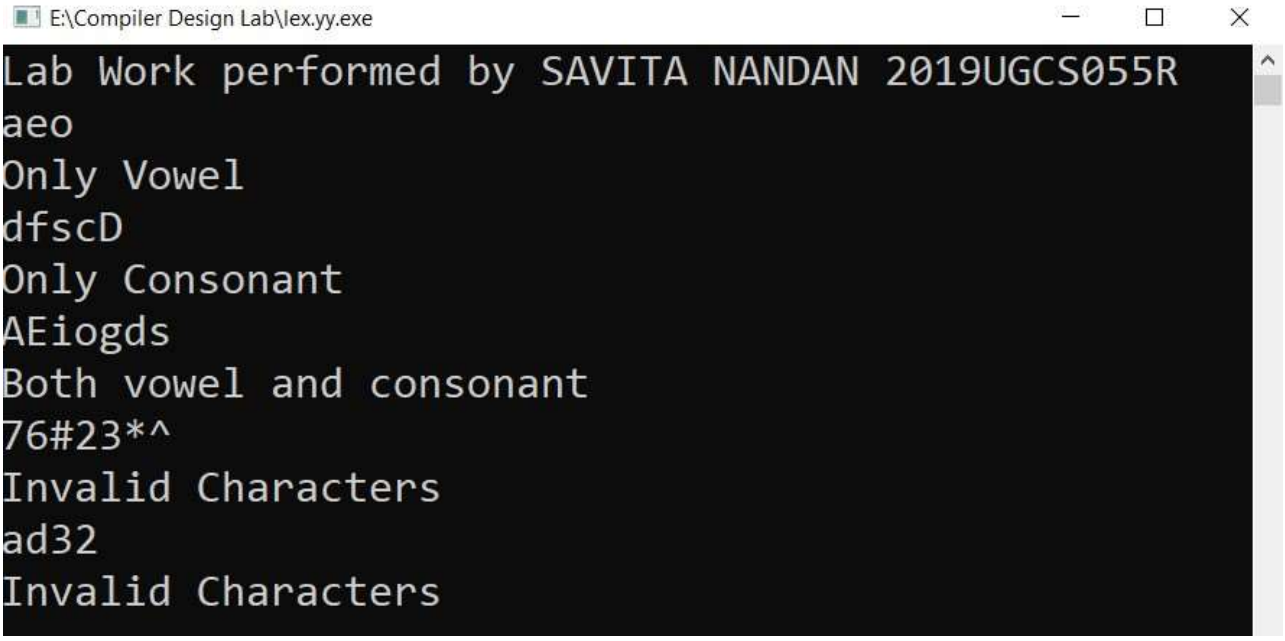
Date – 20.08.2021

Q1. Write a Lex program to identify vowel and consonant.

Code –

```
%{
  //macros of C program
%}
consonant[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z]
vowel[AEIOUaeiou]
mixed[A-Za-z]
%%

{vowel}* {printf("Only Vowel");};
{consonant}* {printf("Only Consonant");};
{mixed}* {printf("Both vowel and consonant");};
.* {printf("Invalid Characters");};
%%

int main() {
   printf("Lab Work performed by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
}

int yywrap(void) {
   return 0;
}
```

Output –



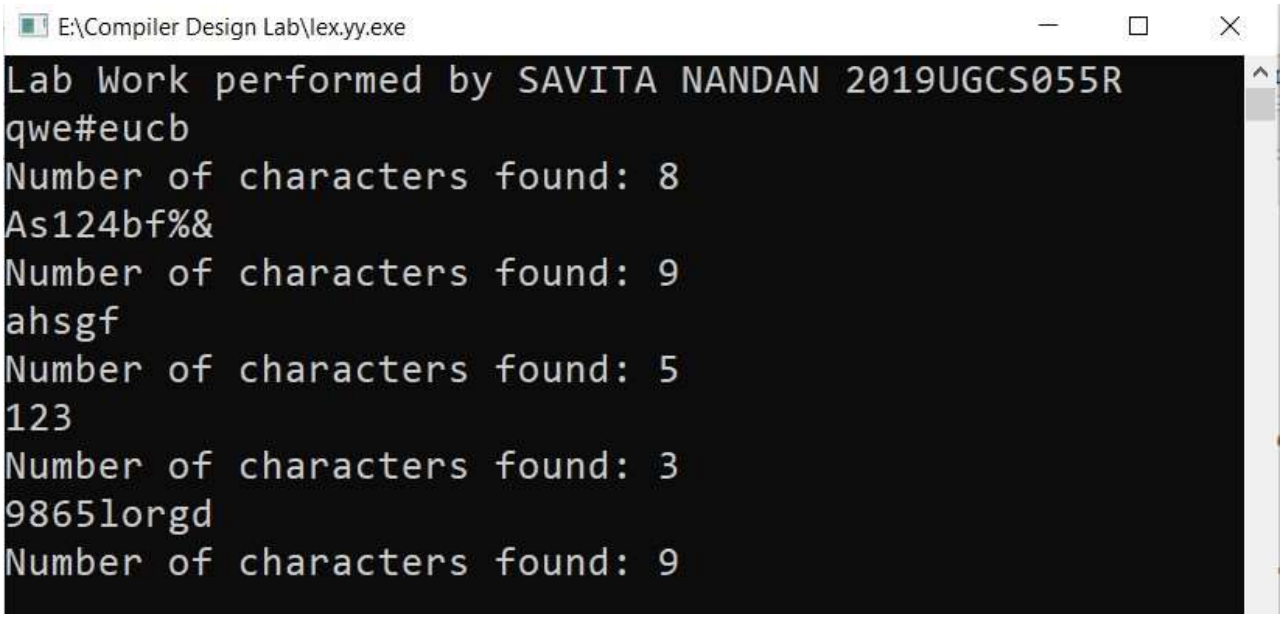
Q2. Write a Lex program to count number of characters in input.

Code –

```
%{
int charcount=0;
%}

%%
. {charcount++;};
\n {printf("Number of characters found: %d\n",charcount); charcount = 0;};
%%

int main() {
   printf("Lab Work performed by SAVITA NANDAN 2019UGCS055R\n");
   yylex();
```

```
}
int yywrap(void) {
   return 0;
}
```

Output –



Q3. Write a Lex program to count the number of vowels, consonant digits and whitespace in the input string.

Code –

```
%{
   int vow_count=0 , const_count=0 , white_space=0 , digit=0;
%}

%%
[aeiouAEIOU] {vow_count++;}
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] {const_count++;}
[0-9] {digit++;}
```

```
[ ]  {white_space++;}
\n {printf("Number of vowels are:  %d\n", vow_count);
    printf("Number of consonants are:  %d\n", const_count);
    printf("Number of digits are:  %d\n", digit);
    printf("Number of white spaces are:  %d\n", white_space);
          vow_count=0,const_count=0,white_space=0,digit=0;};
%%
int yywrap(){}
int main()
{
    printf("Lab Work performed by SAVITA NANDAN 2019UGCS055R\n");
    yylex();
  return 0;
}
```

Output –