NAME: KSHITIJ GUPTA
Enrolment Number: 21162101007
Sub: CD

Practical – 2[Batch-71]

2) Write a lex Program to Scan and Count the number of characters, words, digits, vowels,consonant, special characters and lines in a file.

Code:

```
%{
#include <stdio.h>
int charCount = 0;
int wordCount = 0;
int digitCount = 0;
int vowelCount = 0;
int consonantCount = 0;
int specialCharCount = 0;
int lineCount = 0;
int inWord = 0;  // Flag to keep track of whether we are inside a word
%}
%%
[a-zA-Z] {
  charCount++;
  if (!inWord) {
     wordCount++;
     inWord = 1;
  }
  char ch = yytext[0];
```

```
  if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == 'A' || ch == 'E' ||
ch == 'I' || ch == 'O' || ch == 'U') {
      vowelCount++;
  } else {
      consonantCount++;
  }
}
[0-9] {
  charCount++;
  digitCount++;
  if (!inWord) {
      //wordCount++;
      inWord = 1;
  }
}
[^a-zA-Z0-9\ \n\s] {
  charCount++;
  specialCharCount++;
  if (!inWord) {
      wordCount++;
      inWord = 1;
  }
}
\n {
  charCount++;
  lineCount++;
  inWord = 0;
}
```

```
[ \t]+ {
  inWord = 0;
}
. {
  charCount++;
  if (!inWord) {
    wordCount++;
    inWord = 1;
  }
}
%%
int yywrap() {
  // Return 1 to indicate the end of input
  return 1;
}
int main(int argc, char* argv[]) {
  if (argc < 2) {
    printf("Usage: %s <filename>\n", argv[0]);
    return 1;
  }
  FILE* file = fopen(argv[1], "r");
  if (file == NULL) {
    printf("Error opening file %s\n", argv[1]);
    return 1;
  }
  yyin = file;
  yylex();
  printf("Number of characters: %d\n", charCount);
```

```
    printf("Number of words: %d\n", wordCount);
    printf("Number of digits: %d\n", digitCount);
    printf("Number of vowels: %d\n", vowelCount);
    printf("Number of consonants: %d\n", consonantCount);
    printf("Number of special characters: %d\n", specialCharCount);
    printf("Number of lines: %d\n", lineCount);
    fclose(file);
    return 0;
}
```

Step-1:

```
(c) Microsoft Corporation. All rights reserved.

D:\SEM-7\CD\pr2\PR2_2>flex PR2_2.l
"PR2_2.l", line 50: warning, rule cannot be matched
```

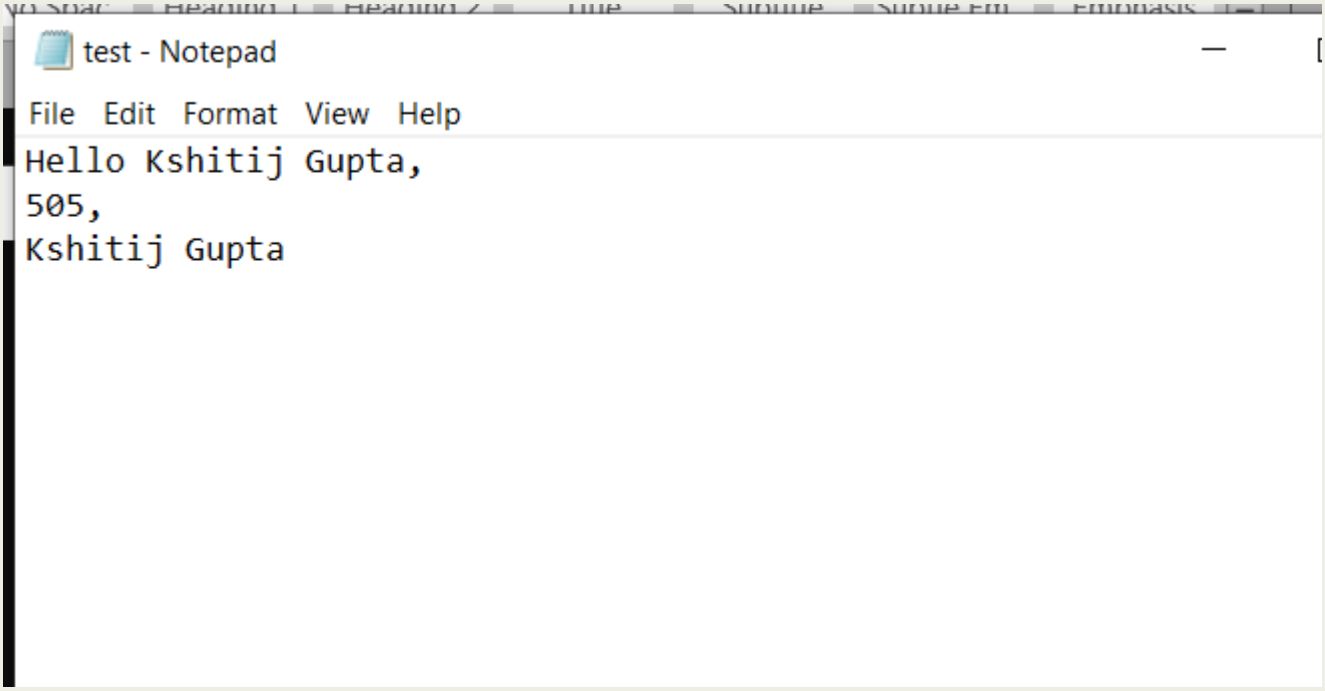
Step-2:

```
D:\SEM-7\CD\pr2\PR2_2>gcc lex.yy.c

D:\SEM-7\CD\pr2\PR2_2>a
Usage: a <test.txt>
```

Step-3:

```
D:\SEM-7\CD\pr2\PR2_2>a
Usage: a <test.txt>

D:\SEM-7\CD\pr2\PR2_2>test.txt

D:\SEM-7\CD\pr2\PR2_2>a test.txt
Number of characters: 36
Number of words: 5
Number of digits: 3
Number of vowels: 10
Number of consonants: 19
Number of special characters: 2
Number of lines: 2

D:\SEM-7\CD\pr2\PR2_2>
```

Step-4:



3) Write a lex Program to recognize regular expression under 'a', 'a*b+', 'abb' , b* over the input set {a,b}.

Code:

```
%{
#include <stdio.h>
%}
%option noyywrap
%%
"b"* {
  printf("String matched: a*b+\n");
  printf("String matched: b*\n");
}
"abb" {
  printf("String matched: abb\n");
}
("a")*("b")+ {
```

```
  printf("String matched: a*b+\n");
}
"a" {
  printf("String matched: a\n");
}
\n {
  // Ignore empty lines
}
. {
  printf("Invalid input\n");
}
%%
int main(int argc, char* argv[]) {
  if (argc != 2) {
      printf("Usage: %s <input_string>\n", argv[0]);
      return 1;
  }
  yy_scan_string(argv[1]);
  yylex();
  if (argv[1][0] == '\0') {
      printf("String matched: b*\n");
  }
  return 0;
}
```

Step-1:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

D:\SEM-7\CD\pr2\PR2_3>flex PR2_3.l

D:\SEM-7\CD\pr2\PR2_3>
```

Step-2:

```
D:\SEM-7\CD\pr2\PR2_3>gcc lex.yy.c

D:\SEM-7\CD\pr2\PR2_3>
```

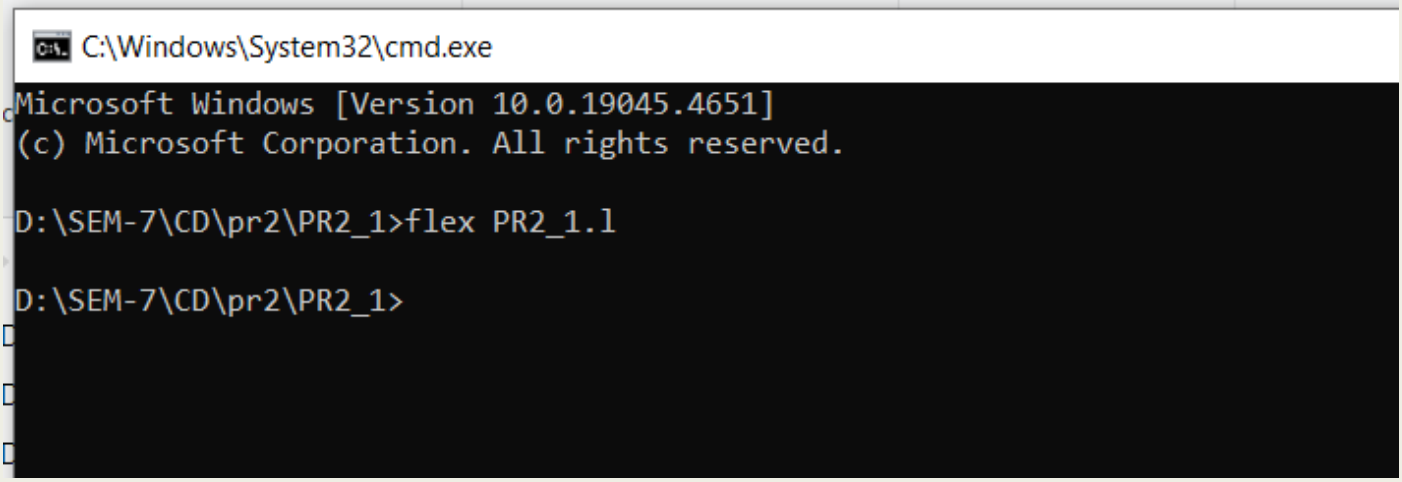
Step-3:

```
D:\SEM-7\CD\pr2\PR2_3>a "aaa"
String matched: a
String matched: a
String matched: a

D:\SEM-7\CD\pr2\PR2_3>a "baaba"
String matched: a*b+
String matched: b*
String matched: a*b+
String matched: a

D:\SEM-7\CD\pr2\PR2_3>a "babbab"
String matched: a*b+
String matched: b*
String matched: abb
String matched: a*b+

D:\SEM-7\CD\pr2\PR2_3>a "kshitij"
Invalid input
Invalid input
Invalid input
Invalid input
Invalid input
Invalid input
Invalid input
```

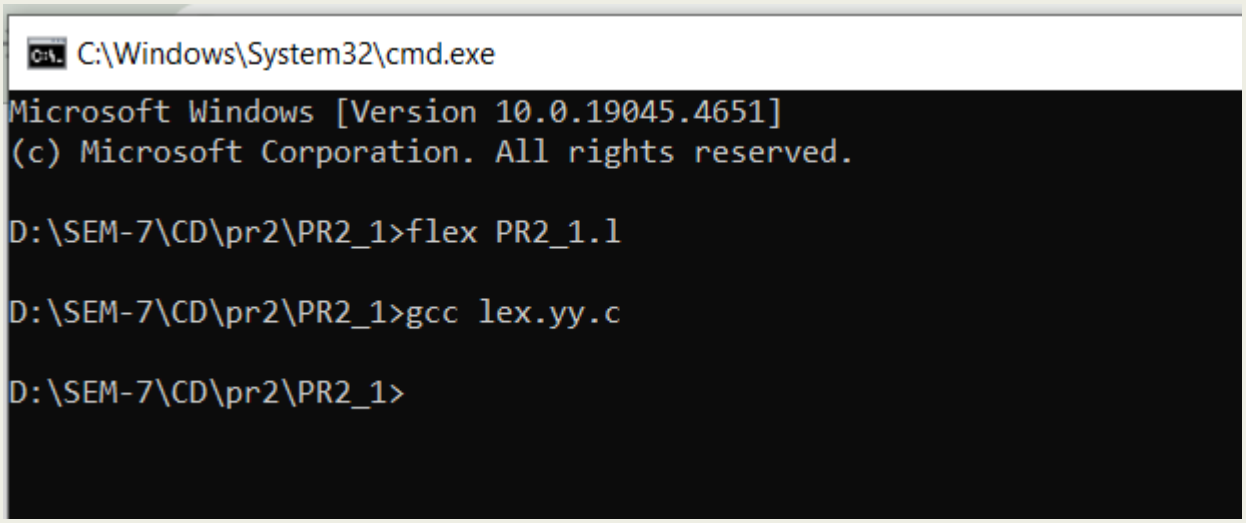
1)lex program to count number of words and digit

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

int word_count = 0;
int digit_count = 0;
%}

%%
[0-9]+      { digit_count += yyleng; }   // Count each digit
[ \t\n]+    { /* Ignore whitespace */ }
[a-zA-Z]+   { word_count++; }           // Count each word
.           { /* Ignore other characters */ }
%%

int main(int argc, char **argv)
{
    yylex();
    printf("Number of words: %d\n", word_count);
    printf("Number of digits: %d\n", digit_count);
    return 0;
}

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

Step-1:



Step-2:



Step-3: