NAME: KSHITIJ GUPTA

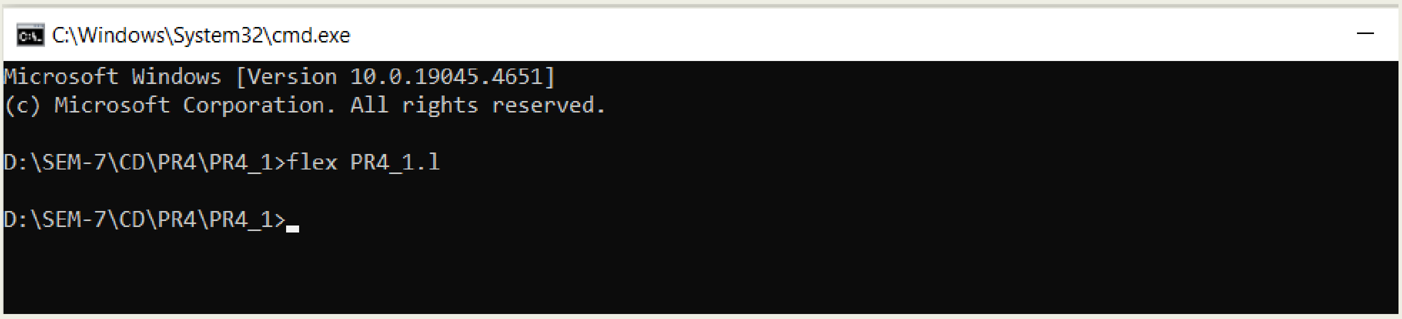
Enrolment Number: 21162101007

Sub: CD

Practical – 4[Batch-71]

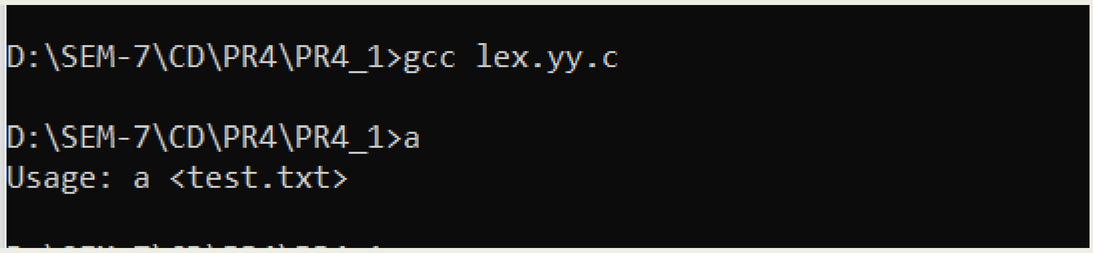
1. to Identify integer, Float and Exponential numbers

Step-1:



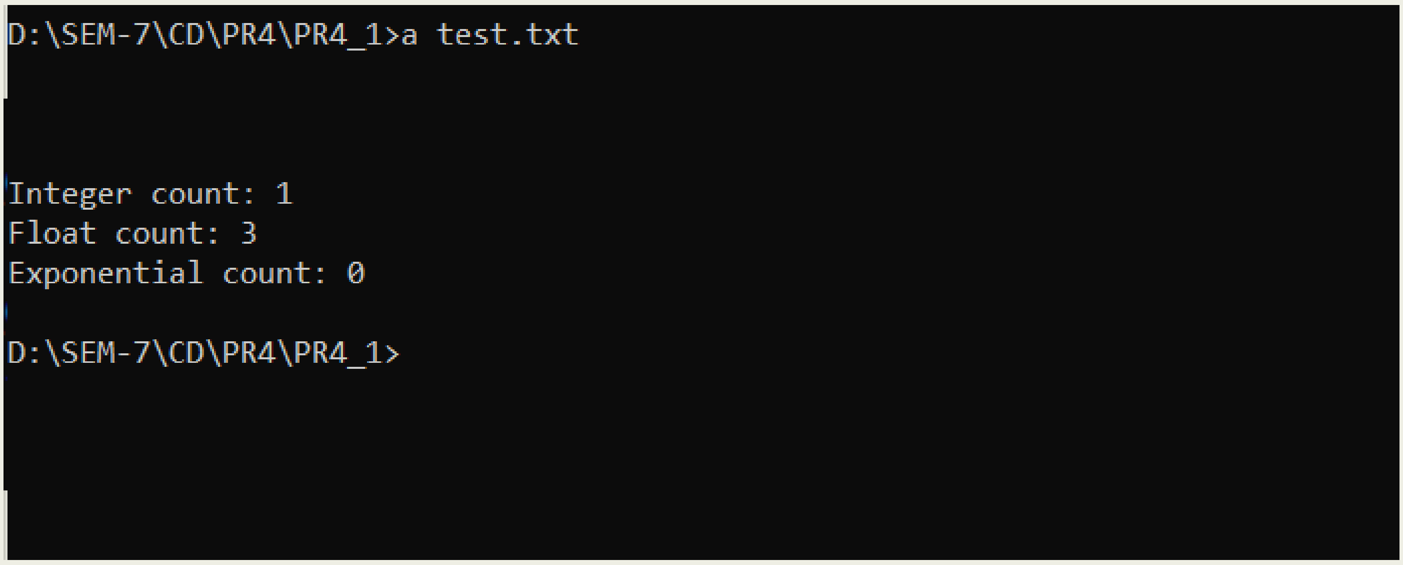
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

D:\SEM-7\CD\PR4\PR4_1>flex PR4_1.l

D:\SEM-7\CD\PR4\PR4_1>
```

Step-2:



```
D:\SEM-7\CD\PR4\PR4_1>gcc lex.yy.c

D:\SEM-7\CD\PR4\PR4_1>a
Usage: a <test.txt>
```

Step-3:



```
D:\SEM-7\CD\PR4\PR4_1>a test.txt



Integer count: 1
Float count: 3
Exponential count: 0

D:\SEM-7\CD\PR4\PR4_1>
```

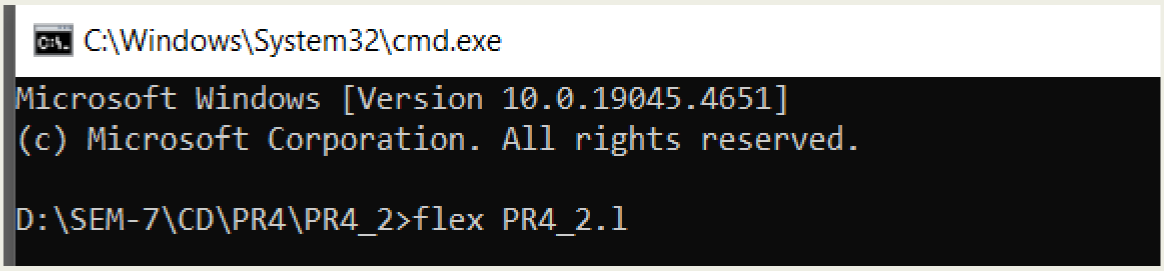
Code:

```
%{

#include <stdio.h>
int int_count = 0;
int float_count = 0;
int exp_count = 0;

%}

%%

[+-]?[0-9]+                   { int_count++; }
[+-]?[0-9]*\.[0-9]+([0-9]+)? { float_count++; }
[+-]?[0-9]+(\.[0-9]*)?[eE][+-]?[0-9]+ { exp_count++; }

%%

int yywrap() {
  // Return 1 to indicate the end of input
  return 1;
}

int main(int argc, char* argv[]) {
   if (argc != 2) {
      printf("Usage: %s <test.txt>\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("Integer count: %d\n", int_count);
    printf("Float count: %d\n", float_count);
    printf("Exponential count: %d\n", exp_count);

    fclose(file);
    return 0;
}
```

2. Identify Single and Multiline comments in C program

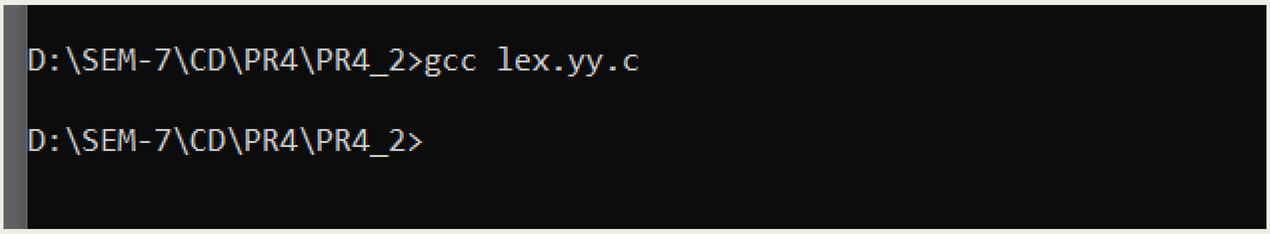
Step-1:



C:\Windows\System32\cmd.exe

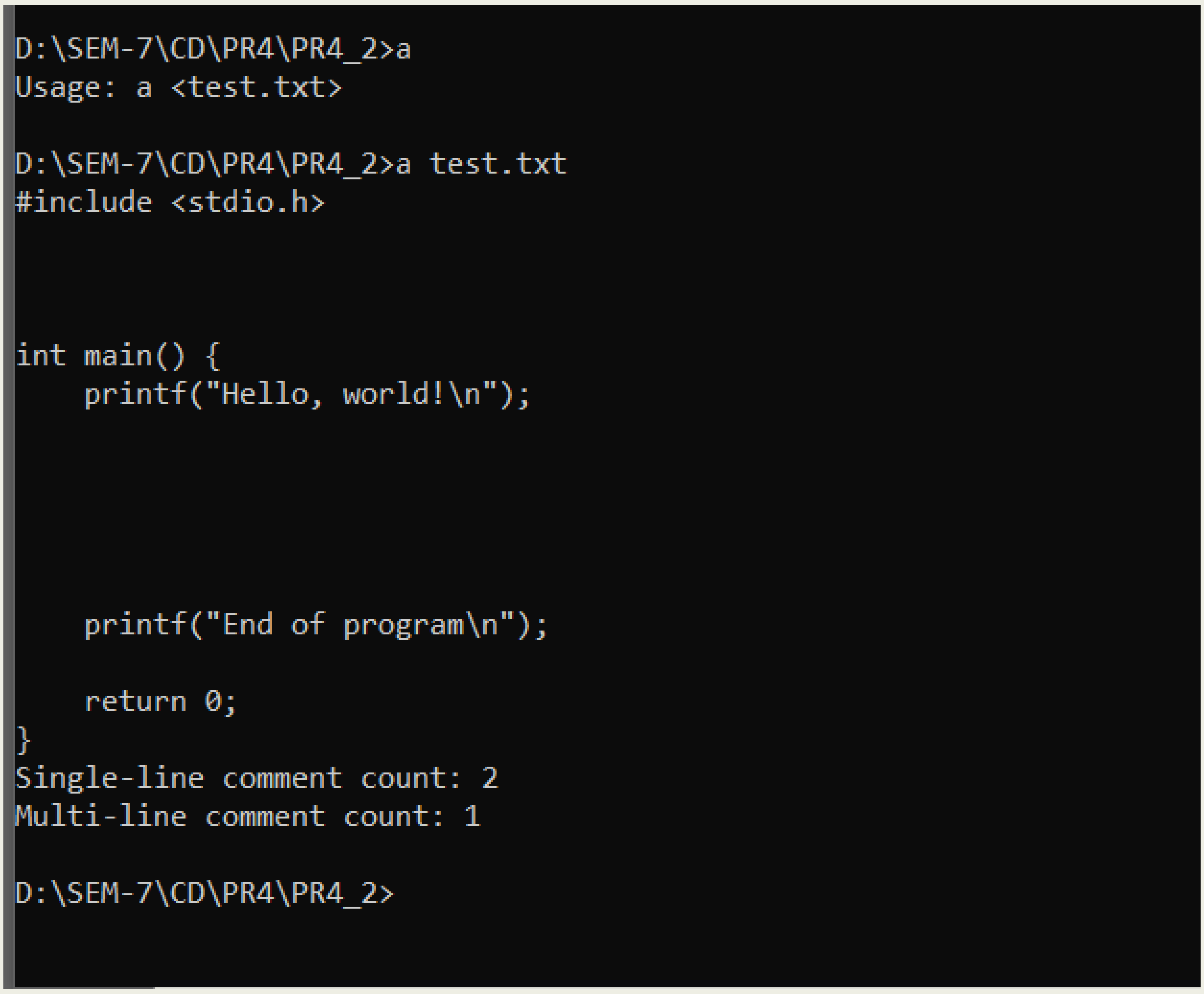
```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

D:\SEM-7\CD\PR4\PR4_2>flex PR4_2.l
```

Step-2:



```
D:\SEM-7\CD\PR4\PR4_2>gcc lex.yy.c

D:\SEM-7\CD\PR4\PR4_2>
```

Step-3:



```
D:\SEM-7\CD\PR4\PR4_2>a
Usage: a <test.txt>

D:\SEM-7\CD\PR4\PR4_2>a test.txt
#include <stdio.h>



int main() {
    printf("Hello, world!\n");





    printf("End of program\n");

    return 0;
}
Single-line comment count: 2
Multi-line comment count: 1

D:\SEM-7\CD\PR4\PR4_2>
```

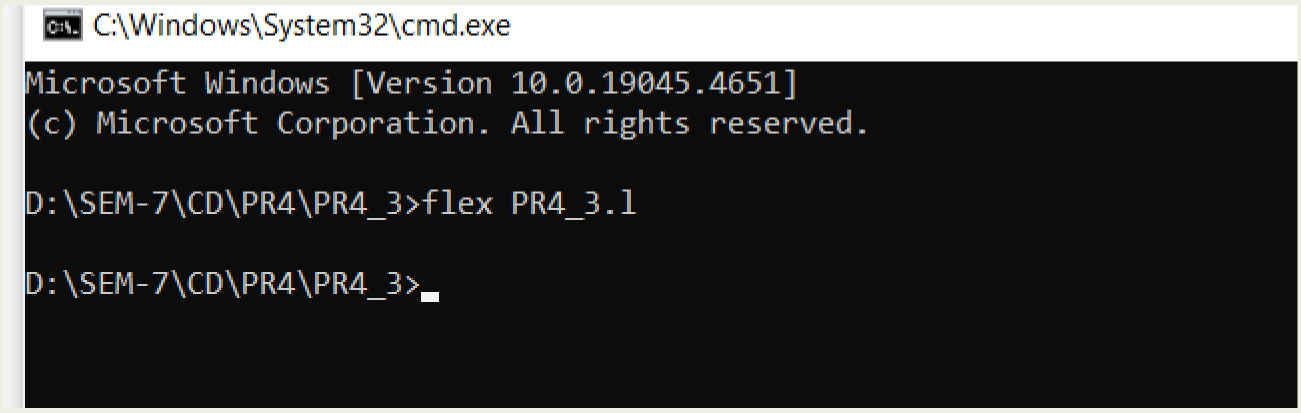
Code:

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

int single_line_comment_count = 0;
int multi_line_comment_count = 0;

%}

%%

"//".*                          { single_line_comment_count++; }
"/*"([^*]|\*+[^/])*\*+"/"       { multi_line_comment_count++; }

%%

int yywrap() {
  // Return 1 to indicate the end of input
  return 1;
}

int main(int argc, char* argv[]) {
   if (argc != 2) {
      printf("Usage: %s <test.txt>\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("Single-line comment count: %d\n", single_line_comment_count);
    printf("Multi-line comment count: %d\n", multi_line_comment_count);

    fclose(file);
    return 0;
}
```

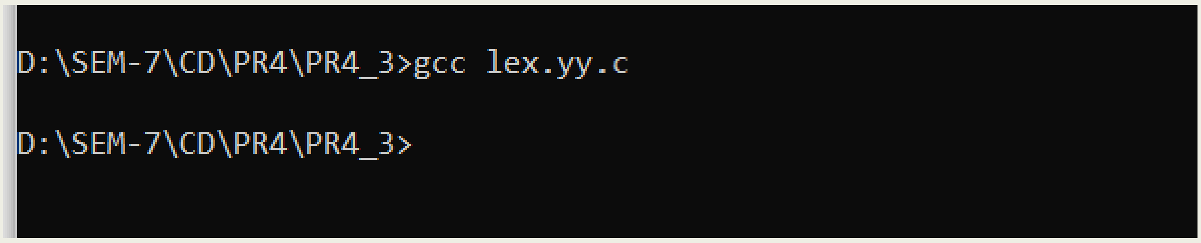
3. **Identify valid tokens in given statement**
scanf("%d %d",&a,&b);
printf("%d %d",a,b);

Step-1:


C:\Windows\System32\cmd.exe

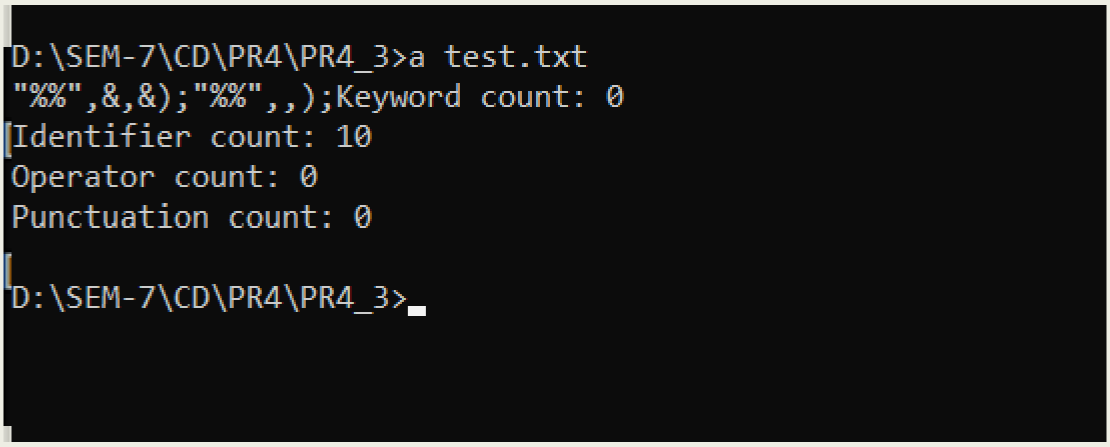
```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

D:\SEM-7\CD\PR4\PR4_3>flex PR4_3.l

D:\SEM-7\CD\PR4\PR4_3>_
```

Step-2:



```
D:\SEM-7\CD\PR4\PR4_3>gcc lex.yy.c

D:\SEM-7\CD\PR4\PR4_3>
```

Step-3:



```
D:\SEM-7\CD\PR4\PR4_3>a test.txt
"%%",&,&);"%%",,);Keyword count: 0
Identifier count: 10
Operator count: 0
Punctuation count: 0

D:\SEM-7\CD\PR4\PR4_3>_
```

Code:

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

int keyword_count = 0;
int identifier_count = 0;
int operator_count = 0;
int punctuation_count = 0;

%}

%%

"scanf" | "printf"            { keyword_count++; }
[a-zA-Z_][a-zA-Z0-9_]*        { identifier_count++; }
"(" | ")" | "," | ";"         { punctuation_count++; }
"=" | "==" | "!=" | "<" | ">" | "<=" | ">=" | "+" | "-" | "*" | "/" | "%" {
operator_count++; }
[ \t\n]+                      ;  // Ignore whitespace

%%

int yywrap() {
   return 1;
}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        printf("Usage: %s <test.txt>\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("Keyword count: %d\n", keyword_count);
    printf("Identifier count: %d\n", identifier_count);
    printf("Operator count: %d\n", operator_count);
    printf("Punctuation count: %d\n", punctuation_count);

    fclose(file);
    return 0;
}
```