System Software Practicals

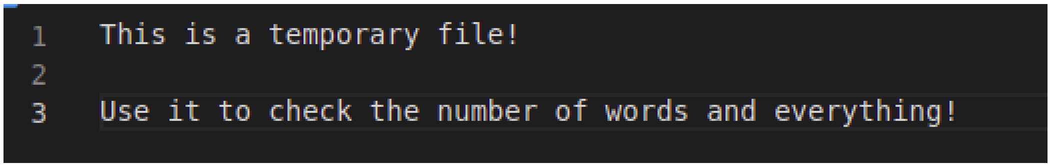
Assignment 6

Krunal Rank
U18CO081

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

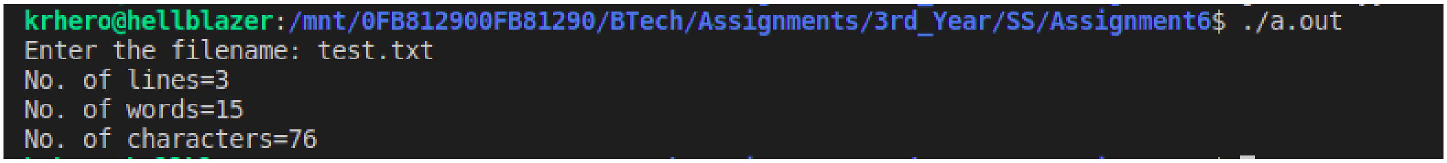
```
%{
#include<stdio.h>
int lines=0,words = 0, characters = 0,cnt=0;
%}
 %%
[a-zA-Z0-9_] {cnt++;characters++;}
" " {if(cnt!=0){words++;cnt=0;}characters++;}
[\t] words++;
[\n] {words++;lines++;}
%%
 int yywrap(){}
int main(){

FILE *fp;
char filename[50];
printf("Enter the filename: ");
scanf("%s",filename);
fp = fopen(filename,"r");
yyin = fp;
 yylex();

printf("No. of lines=%d\n", lines);
printf("No. of words=%d\n", words);
printf("No. of characters=%d\n", characters);
 return 0;
}
```

Input:



```
This is a temporary file!

Use it to check the number of words and everything!
```

Output:



```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
Enter the filename: test.txt
No. of lines=3
No. of words=15
No. of characters=76
```

2. Design a scanner to

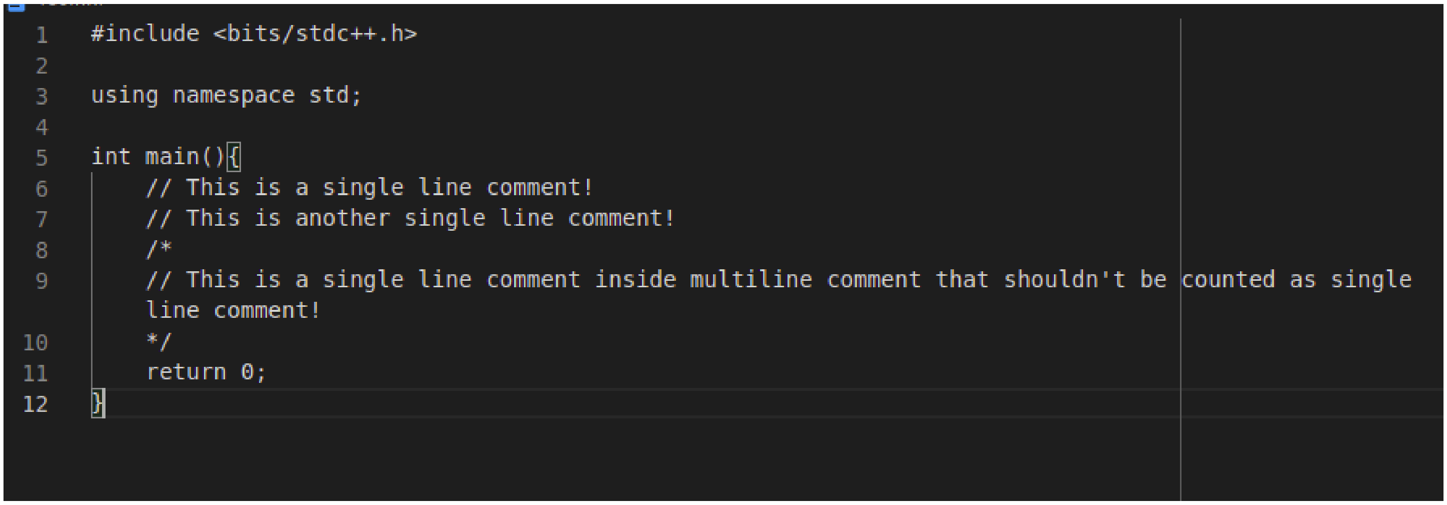
(a) Count number of single and multiple line comments from a C program available in xyz.txt file. [Note: You can create any txt file having sample C code which contains single and multiple line comments]

(b) Remove comment lines from the C program.

```
%{
#include<stdio.h>
int flag=0,single=0,multi=0;
%}
 %%
"/*" { comment(); multi++;}
"//".* { single++;}
%%
 int yywrap(){}
void comment()
{
   int c;

   while ((c = input()) != 0)
       if (c == '*')
       {
           while ((c = input()) == '*')
               ;

           if (c == '/')
               return;

           if (c == 0)
               break;
       }
}
int main(){

FILE *fp;
char filename[50];
printf("Enter the filename: ");
scanf("%s",filename);
fp = fopen(filename,"r");
yyin = fp;
yyout = fopen("out.c","w");
 yylex();

printf("No. of Single Line Comments=%d\n", single);
printf("No. of Multi Line Comments=%d\n", multi);
```

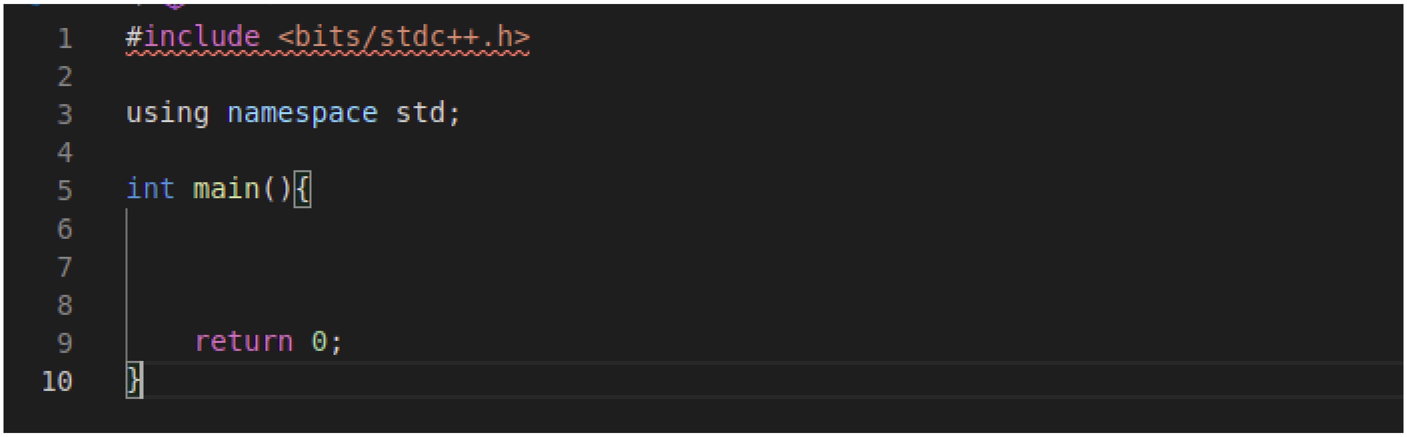
```
 return 0;
}
```

Input:

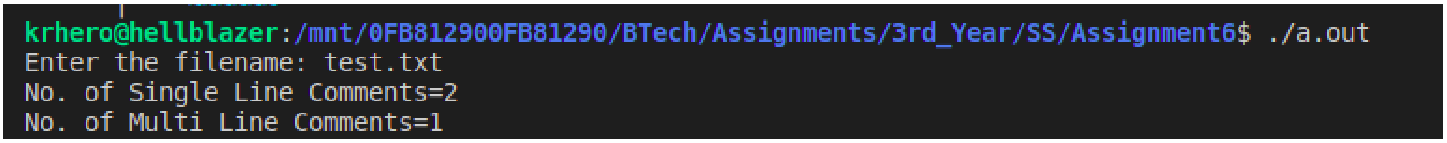


```cpp
#include <bits/stdc++.h>

using namespace std;

int main(){
    // This is a single line comment!
    // This is another single line comment!
    /*
    // This is a single line comment inside multiline comment that shouldn't be counted as single line comment!
    */
    return 0;
}
```

Output:



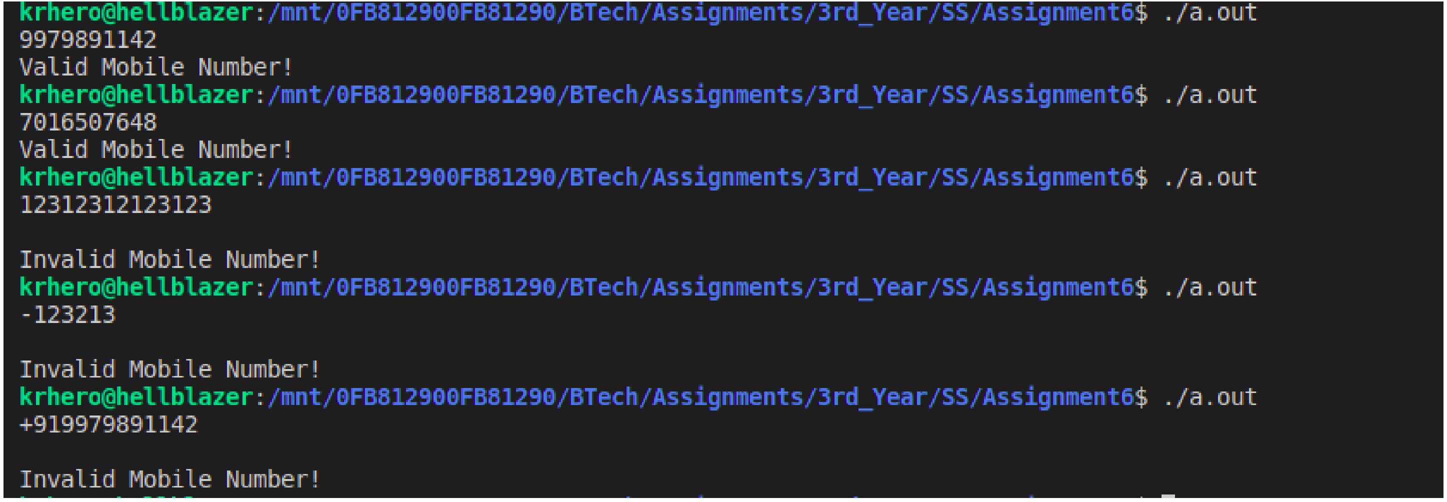
```cpp
#include <bits/stdc++.h>

using namespace std;

int main(){



    return 0;
}
```



```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
Enter the filename: test.txt
No. of Single Line Comments=2
No. of Multi Line Comments=1
```

3. Write a Lex program to check valid/invalid
(a) Mobile number (considering 10-digit mobile number followed by country code +91)
(b) Email address
A:

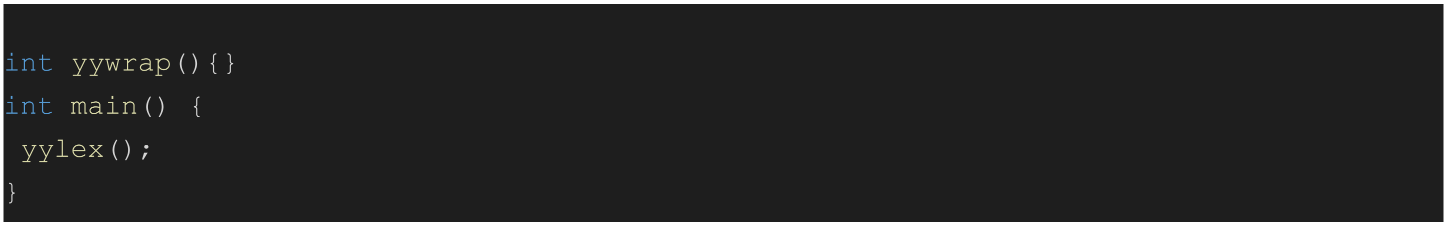
```
%{
#include<stdio.h>
int flag=0,single=0,multi=0;
%}
 %%
[1-9][0-9]{9} {printf("Valid Mobile Number!\n");}
 .+ {printf("Invalid Mobile Number!\n");}
"\n" {return 0;}
%%
 int yywrap(){}
int main(){


yylex();
 return 0;
}
```



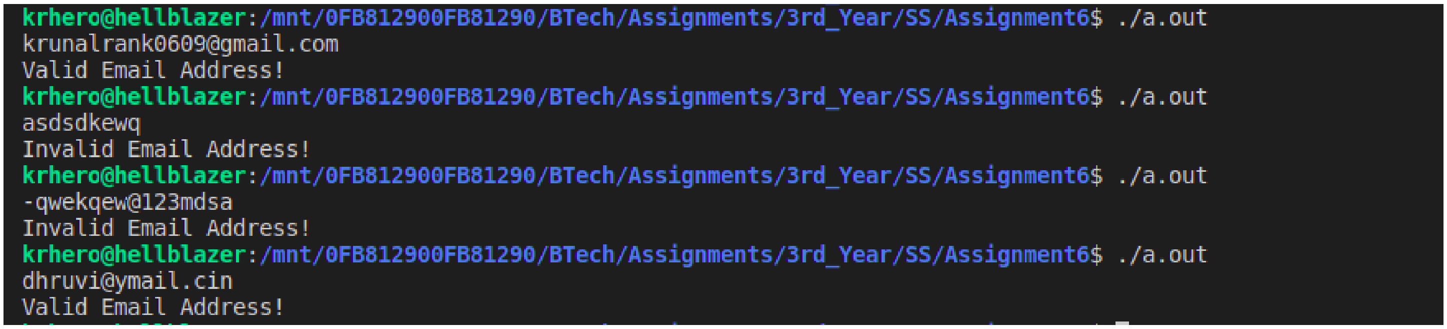
```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
9979891142
Valid Mobile Number!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
7016507648
Valid Mobile Number!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
12312312123123

Invalid Mobile Number!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
-123213

Invalid Mobile Number!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
+919979891142

Invalid Mobile Number!
```

B:

```
/*lex code to accept a  valid email */
%{
#include<stdio.h>
%}

%%

[0-9.a-z]{2,}@[a-z]{4,}(\.[a-z]{2,})+ {printf("Valid Email Address!\n");}
.+ {printf("Invalid Email Address!\n");}
"\n" {return 0;}
%%
```



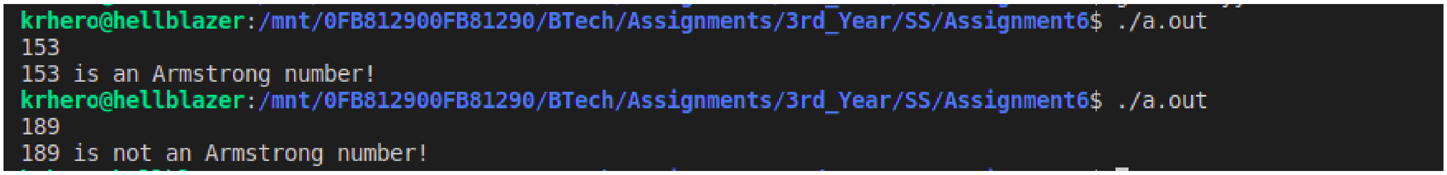
```
int yywrap(){}
int main() {
 yylex();
}
```



```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
krunalrank0609@gmail.com
Valid Email Address!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
asdsdkewq
Invalid Email Address!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
-qwekqew@123mdsa
Invalid Email Address!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
dhruvi@ymail.cin
Valid Email Address!
```

4. Design a scanner to check whether a number is an Armstrong number or not.

```
%{
#include <stdio.h>
#include <math.h>
#include <string.h>
void check(char*);
%}
 %%
[0-9]+ {check(yytext);}
"\n" {return 0;}
%%
int yywrap(){};
int main()
{
   yylex();
    return 0;
}
void check(char* a)
{
   int len = strlen(a), i, num = 0;
   for (i = 0; i < len; i++)
       num = num * 10 + (a[i] - '0');
    int x = 0, y = 0, temp = num;
   while (num > 0) {
       y = (int)pow((double)(num % 10),(double) len);
       x = x + y;
       num = num / 10;
   }
    if (x == temp)
       printf("%d is an Armstrong number!\n", temp);
   else
       printf("%d is not an Armstrong number!\n", temp);
}
```



```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
153
153 is an Armstrong number!
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/SS/Assignment6$ ./a.out
189
189 is not an Armstrong number!
```