

Compiler Design Lab 1

Write a C program to :

1. To count the number of lines and characters in a file.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>

int main(){
    int countline=0,countc=0;
    char c;FILE *fptr;
    fptr= fopen("input.txt","r");
    if(!fptr){
        printf("Cannot open file\n");
        exit(0);
    }
    c=getc(fptr);
    while(c!=EOF){
        if(c=='\n') countline++;
        /*Assuming that whitespace and new line are not considered as characters*/
        else countc++;

        c=getc(fptr);
    }
    if(countc >0) countline++;
    printf("Total number of lines and characters is %d and %d respectively\n",countline,countc);
    return 0;
}
```

We also create an “input.txt” file for testing and the following along with the output is shown as following:

Output:

```
ugcse@pglab-cp: ~/Desktop/AyushGoyal_CDLab/Lab_1
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ touch input.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat > input.txt
This is a CD test text file.
I am Ayush Goyal.
Today is a tuesday.ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat input.txt
This is a CD test text file.
I am Ayush Goyal.
Today is a tuesday.ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat input.tx
t
t: command not found
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ gcc l1q1.c -o l1q1
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ ./l1q1
Total number of lines and characters is 3 and 64 respectively
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
```

2. To reverse the file contents and store in another file. Also display the size of file using file handling function.

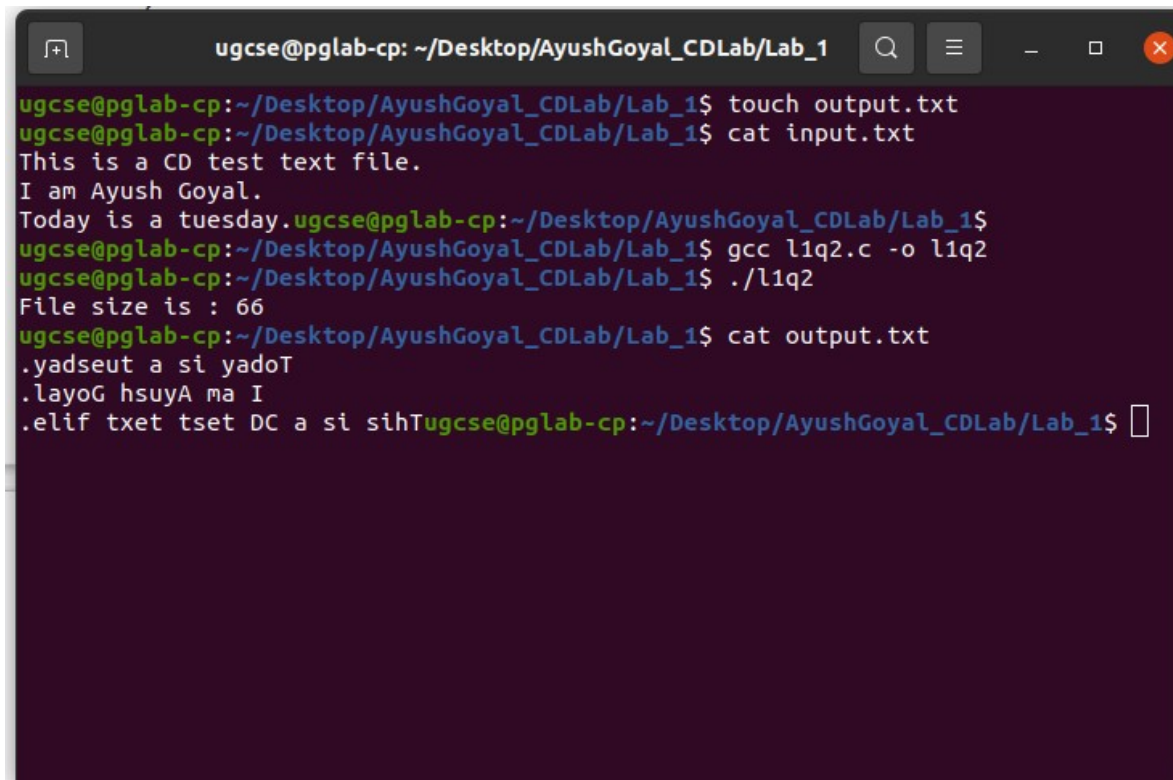
Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <sys/stat.h>

int main(){
    FILE *fa,*fb;
    char ch,buffer[1024];
    int i=0;
    fa=fopen("input.txt","r");
    fb=fopen("output.txt","w+");
    if(!fa||!fb){
        printf("Cannot open the file");
        exit(1);
    }
    while(ch!=EOF){
        ch=getc(fa);
        buffer[i++]=ch;
    }
    for(int j=(i-2);j>=0;j--){
        ch=buffer[j];
        fputc(ch,fb);
    }
    fseek(fa,0,SEEK_END);
    int size=ftell(fa);
    printf("File size is : %d\n",size );
}
```

We create an output.txt file and run the following commands to carry out the operations required:

Output:

A terminal window with a dark purple background. The title bar shows the user 'ugcse@pglab-cp' and the directory '~/Desktop/AyushGoyal_CDLab/Lab_1'. The terminal contains the following commands and output:

```
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ touch output.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat input.txt
This is a CD test text file.
I am Ayush Goyal.
Today is a tuesday.ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ gcc l1q2.c -o l1q2
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ ./l1q2
File size is : 66
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat output.txt
.yadseut a si yadoT
.layoG hsuyA ma I
.elif txet tset DC a si sihTugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
```

3. That merges lines alternatively from 2 files and stores it in a resultant file.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <sys/stat.h>

int main(){
    FILE *f1,*f2,*f3;
    char c1,c2;
    f1=fopen("file1.txt","r");
    f2=fopen("file2.txt","r");
    f3=fopen("output.txt","w+");
    if(!f1||!f2||!f3){
        printf("Cannot open file\n");
        exit(1);
    }
    while(1){
```

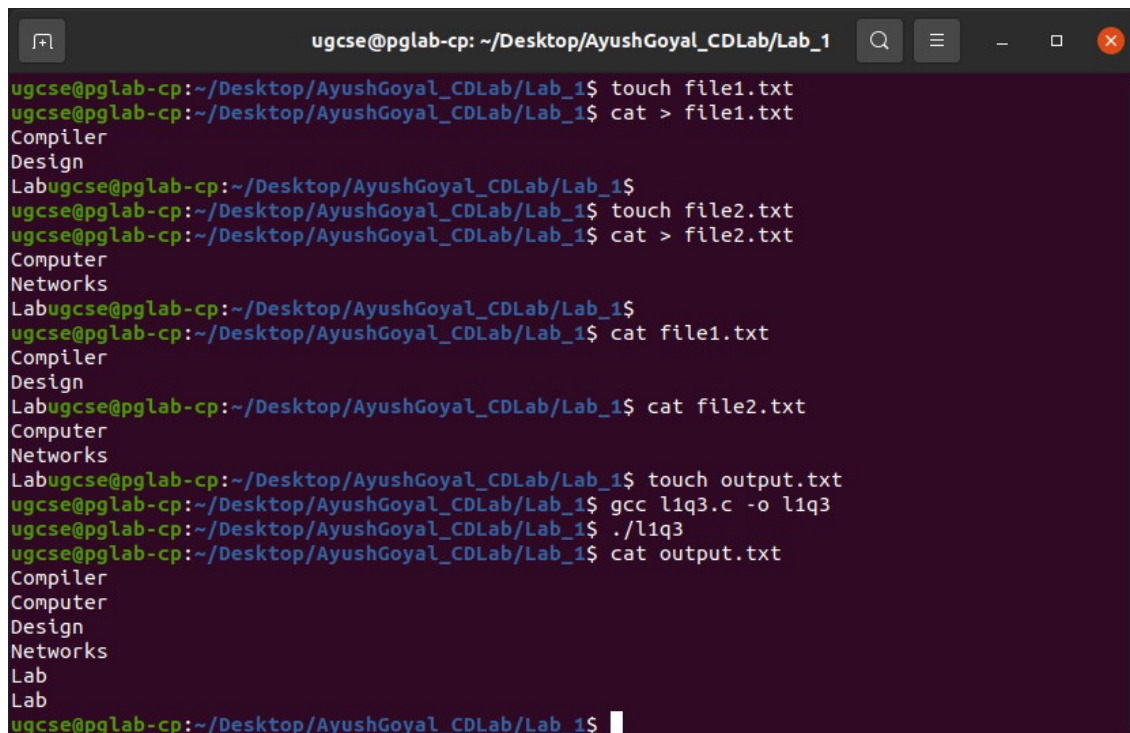
```

        if(c1!=EOF){
            c1=fgetc(f1);
            while(c1!='\n'){
                if(c1==EOF) break;
                fputc(c1,f3);
                c1=fgetc(f1);
            }
            fputc('\n',f3);
        }
        if(c2!=EOF){
            c2=fgetc(f2);
            while(c2!='\n'){
                if(c2==EOF) break;
                fputc(c2,f3);
                c2=fgetc(f2);
            }
            fputc('\n',f3);
        }
        if(c1==EOF && c2==EOF) break;
    }
    return 0;
}

```

We create two files with sensible texts, namely file1.txt and file2.txt. Then we compile and run our C program and see the required operation in the file output.txt :

Output :



```

ugcse@pglab-cp: ~/Desktop/AyushGoyal_CDLab/Lab_1
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ touch file1.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat > file1.txt
Compiler
Design
Labugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ touch file2.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat > file2.txt
Computer
Networks
Labugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat file1.txt
Compiler
Design
Labugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat file2.txt
Computer
Networks
Labugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ touch output.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ gcc l1q3.c -o l1q3
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ ./l1q3
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$ cat output.txt
Compiler
Computer
Design
Networks
Lab
Lab
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_1$

```

THE END