CD Lab 2

Name: Rhea Adhikari
Reg No: 190905156
Section D
Roll No:23

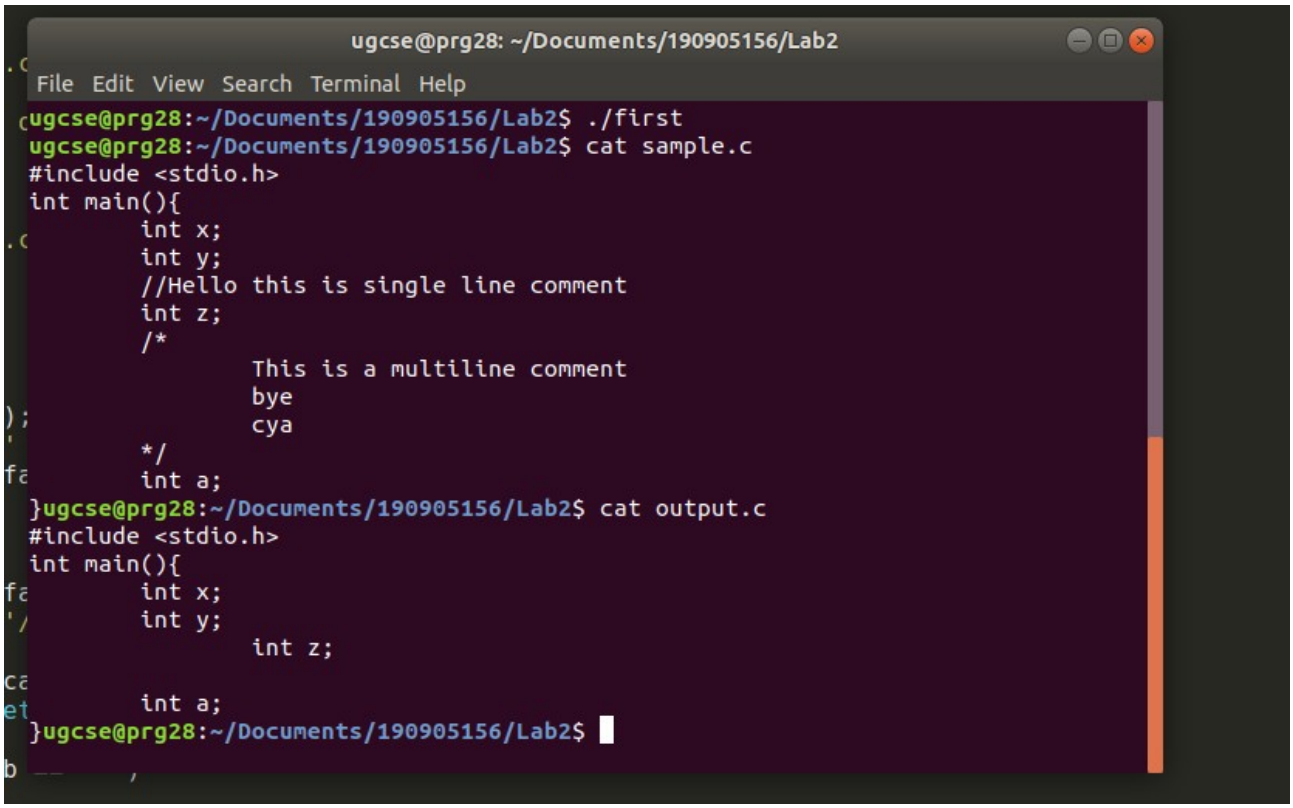
Solved Exercise

Q) Program to remove single and multiline comments from a given 'C' file.

```
#include <stdio.h>
int main()
{
	FILE *fa, *fb;
	int ca, cb;
	fa = fopen("sample.c", "r");
	if (fa == NULL){
		printf("Cannot open file \n");
		exit(0);
	}

	fb = fopen("output.c", "w");
	ca = getc(fa);
	while (ca != EOF)
	{
		if(ca==' ')
		{
			putc(ca,fb);
			while(ca==' ')
			ca = getc(fa);
		}
		if (ca=='/')
		{
			cb = getc(fa);
			if (cb == '/')
			{
				while(ca != '\n')
				ca = getc(fa);
			}
			else if (cb == '*')
			{
				do
				{
					while(ca != '*')
					ca = getc(fa);
					ca = getc(fa);
				} while (ca != '/');
			}
			else
			{
				putc(ca,fb);
				putc(cb,fb);
```

```
                }
        }
        else putc(ca,fb);
        ca = getc(fa);
    }
    fclose(fa);
    fclose(fb);
    return 0;
}
```

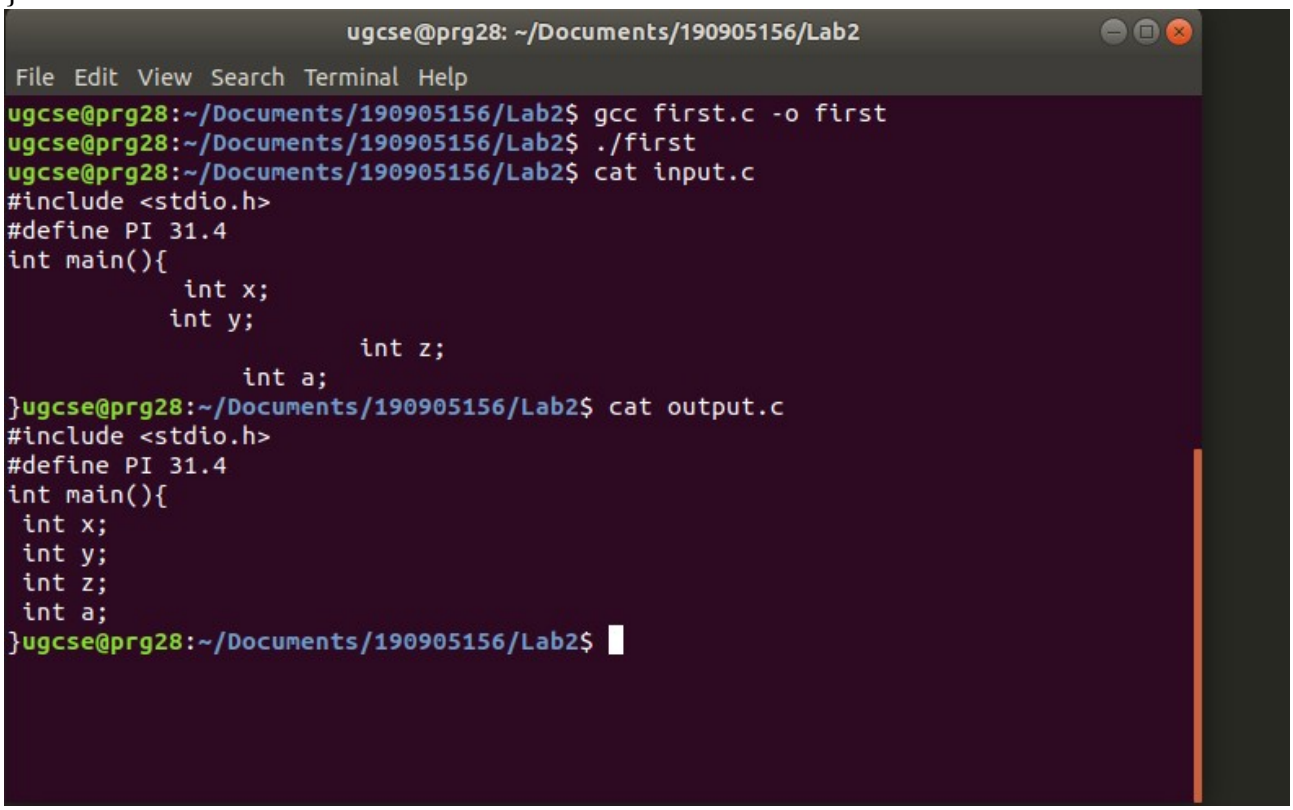


Q1) That takes a file as input and replaces blank spaces and tabs by single space and writes the output to a file.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char c1,c2;
    FILE *f1,*f2;
    f1= fopen("input.c","r");
    f2= fopen("output.c","w");
    if(f1 == NULL || f2 ==NULL)
    {
        printf("Input or the Output file does not exist \n");
        return 1;
    }
    c1=fgetc(f1);
    while(c1 != EOF)
    {
        if(c1 == '/')
```

```
{
        c2 = getc(f1);
        if(c2 == '/')
        {
                putc(c1,f2);
                putc(c2,f2);
                c1 = getc(f1);
                while(c1 !='\n')
                {
                putc(c1,f2);
                c1 = getc(f1);
                }
        }
        else if(c2 == '*')
        {
                putc(c1,f2);
                putc(c2,f2);
                c1 = getc(f1);
                do
                {
                while(c1 != '*')
                {
                        putc(c1,f2);
                        c1 = getc(f1);
                }
                putc(c1,f2);
                c1 = getc(f1);
                } while(c1 != '/');
        }
}
if(c1 == "")
{
        putc(c1,f2);
        c1 = getc(f1);
        while(c1 != "")
        {
                putc(c1,f2);
                c1 = getc(f1);
        }
        putc(c1,f2);
        c1 = getc(f1);
}
if(c1 == ' ' || c1 == '\t')
{
        putc(' ',f2);
        while(c1 == ' ' || c1 == '\t')
        {
        c1 = getc(f1);
        }
}
putc(c1,f2);
c1 = getc(f1);
```

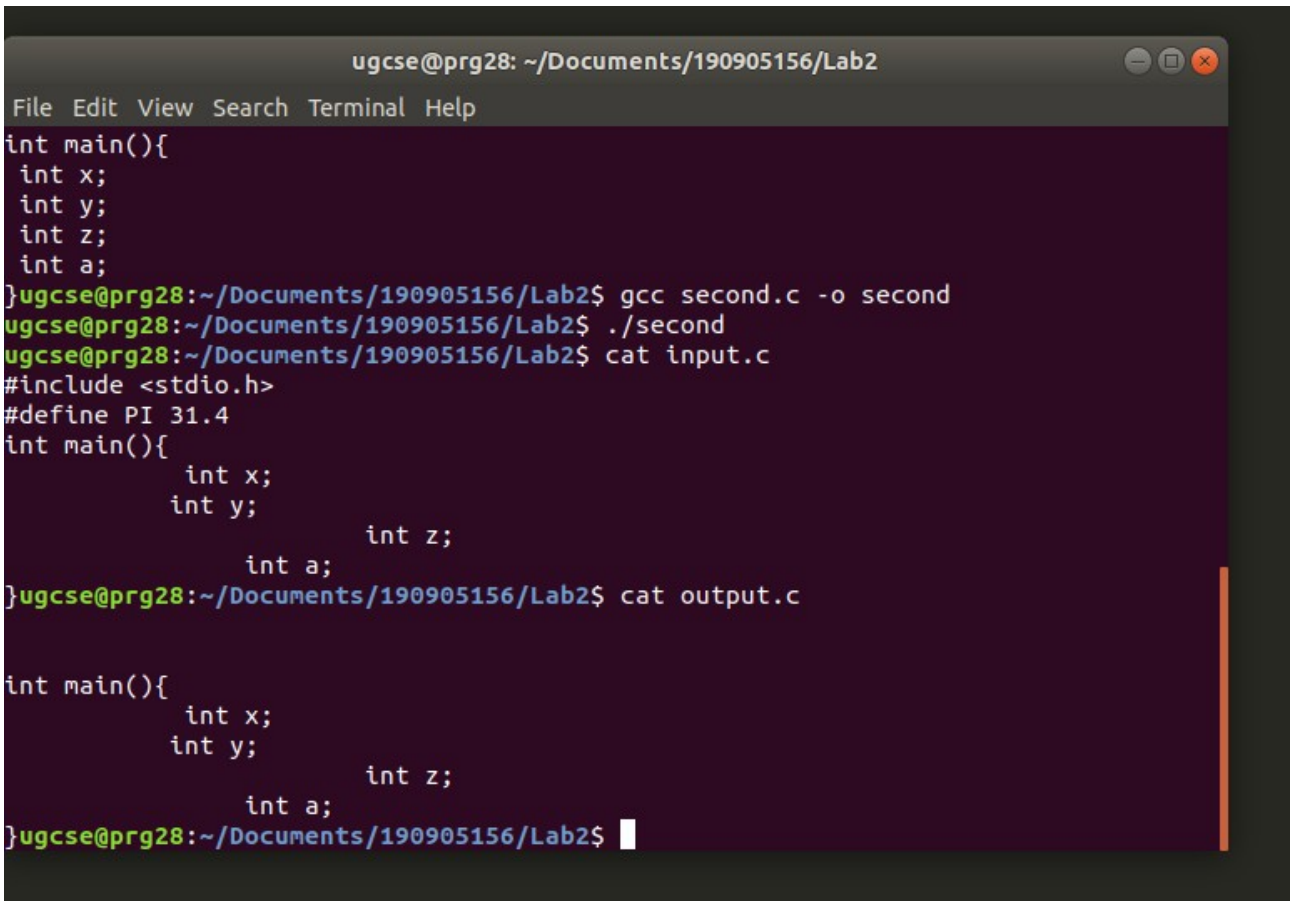
```
        }
        fclose(f1);
        fclose(f2);
        return 0;
}
```



Q 2) To discard preprocessor directives from the given input ‘C’ file.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
        char c1,c2;
        FILE *f1,*f2;
        f1= fopen("input.c","r");
        f2= fopen("output.c","w");
        if(f1 == NULL || f2 ==NULL)
        {
                printf("Input or the Output file does not exist \n");
                return 1;
        }
        c1=fgetc(f1);
        while(c1 != EOF)
        {
                if(c1 == '/')
                {
                        c2 = getc(f1);
                        if(c2 == '/')
                        {
```

```
                        putc(c1,f2);
                        putc(c2,f2);
                        c1 = getc(f1);
                        while(c1 !='\n')
                        {
                                putc(c1,f2);
                                c1 = getc(f1);
                        }
                }
                else if(c2 == '*')
                {
                        putc(c1,f2);
                        putc(c2,f2);
                        c1 = getc(f1);
                        do
                        {
                                while(c1 != '*')
                                {
                                        putc(c1,f2);
                                        c1 = getc(f1);
                                }
                                putc(c1,f2);
                                c1 = getc(f1);
                        } while(c1 != '/');
                }
        }
        if(c1 == "")
        {
                putc(c1,f2);
                c1 = getc(f1);
                while(c1 != "")
                {
                        putc(c1,f2);
                        c1 = getc(f1);
                }
                putc(c1,f2);
                c1 = getc(f1);
        }
        if(c1 == '#')
        {
                while(c1 != '\n')
                c1 = getc(f1);
        }
        putc(c1,f2);
        c1 = getc(f1);
    }
    fclose(f1);
    fclose(f2);
    return 0;
}
```



Q3) That takes C program as input, recognizes all the keywords and prints them in upper case.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
int main()
{
	char keys[10][20] = {"float","printf","scanf","for","char","if","else","do","while","int"};
	char buffer[100];
	FILE *f1;
	int c1,bufferCounter = 0;
	f1 = fopen("input.c","r");
	if(f1 == NULL)
	{
		printf("Cannot open file\n");
		exit(0);
	}
	c1 = getc(f1);
	while(c1 != EOF)
	{
		if(isalpha(c1))
		{
			buffer[bufferCounter++] = c1;
		}
```

```
                else
                {
                        buffer[bufferCounter] = '\0';
                        bufferCounter = 0;
                        for(int i = 0 ; i < 10 ; i++)
                        {
                                if(strcmp(keys[i],buffer) == 0)
                                {
                                        int j = 0;
                                        while(buffer[j])
                                        {
                                                putchar(toupper(buffer[j]));
                                                j++;
                                        }
                                        printf("\n");
                                        break;
                                }
                        }
                }
                c1 = getc(f1);
        }
        fclose(f1);
        return 0;
}
```