Name: Sapthami Upadhya
Section: CSE A
Reg. No.: 230905090

Week 1

Q1. To count the number of lines and characters in a file.

Code:

```
//Write a ‘C’ program to count the number of lines and characters in a file.
#include <stdio.h>
#include <stdlib.h>
int main(){
	FILE *fptr;
	char filename[100], c;
	int lines = 0, chars = 0;
	printf("Enter the filename to open for reading: ");
	scanf("%s", filename);
	fptr = fopen(filename, "r");
	if(fptr == NULL){
		printf("Cannot open file %s\n", filename);
		exit(0);
	}
	c = fgetc(fptr);
	if(c == '\n') lines++;
	if(c != EOF) chars++;
	while(c != EOF){
		c = fgetc(fptr);
		if(c == '\n') lines++;
		if(c != EOF) chars++;
	}
	printf("\nContents read from %s\nTotal characters: %d\nTotal lines: %d\n", filename, chars, lines);
	fclose(fptr);
	return 0;
}
```

Input/Output:

```
Enter the filename to open for reading: source.txt

Contents read from source.txt
Total characters: 525
Total lines: 6
```

Contents of source.txt:

```
The w+ mode is designed for scenarios where you want to create a new file or completely replace the contents of an existing one.

    If the file exists: Its entire content is immediately truncated (deleted), and the file pointer is positioned at the beginning.
    If the file does not exist: A new, empty file is created.
    Pointer Position: The file pointer starts at the beginning of the file.
```

Use Case: Ideal for writing new data from scratch and then potentially reading that data back within the same session.

Q2. To reverse the file contents and store in another file. Also display the size of file using file handling function.

Code:

```
//Write a ‘C’ program to reverse the file contents and store in another file.
//Also display the size of file using file handling function.

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

int main() {
    FILE *fptr1, *fptr2;
    char filename[100];
    long size;
    char c;

    printf("Enter the filename to open for reading: ");
    scanf("%s", filename);
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL) {
        printf("Cannot open file %s\n", filename);
        exit(0);
    }

    printf("Enter the filename to open for writing reversed contents: ");
    scanf("%s", filename);
    fptr2 = fopen(filename, "w+");

    fseek(fptr1, 0, SEEK_END);
    size = ftell(fptr1);
    printf("\nFile size: %ld bytes\n", size);

    fseek(fptr1, -1, SEEK_END);

    while (1) {
        c = fgetc(fptr1);
        fputc(c, fptr2);
        if (ftell(fptr1) == 1)
            break;
        fseek(fptr1, -2, SEEK_CUR);
    }

    printf("\nContents copied to %s\n", filename);

    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

Input/Output:

```
Enter the filename to open for reading: source.txt
Enter the filename to open for writing reversed contents: reversed.txt

File size: 525 bytes

Contents copied to reversed.txt
```

Contents of source.txt:

```
The w+ mode is designed for scenarios where you want to create a new file or completely replace the contents of an existing one.

    If the file exists: Its entire content is immediately truncated (deleted), and the file pointer is positioned at the beginning.
    If the file does not exist: A new, empty file is created.
    Pointer Position: The file pointer starts at the beginning of the file.
    Use Case: Ideal for writing new data from scratch and then potentially reading that data back within the same session.
```

Contents of reversed.txt:

```

 .noisses emas eht nihtiw kcab atad taht gnidaer yllaitnetop neht dna hctarcs morf atad wen gnitirw rof laedI :esaC esU
.elif eht fo gninnigeb eht ta strats retniop elif ehT :noitisoP retnioP
.detaerc si elif ytpme ,wen A :tsixe ton seod elif eht fI
.gninnigeb eht ta denoitisop si retniop elif eht dna ,)deteled( detacnurt yletaidemmi si tnetnoc eritne stI :stsixe elif eht fI

 .eno gnitsixe na fo stnetnoc eht ecalper yletelpmoc ro elif wen a etaerc ot tnaw uoy erehw soiranecs rof dengised si edom +w ehT
```

Q3. That merges lines alternatively from 2 files and stores it in a resultant file.

Code:

```
//Write a ‘C’ program that merges lines alternatively from 2 files and stores it in a resultant file.
#include <stdio.h>
#include <stdlib.h>

int main() {
    FILE *fptr1, *fptr2, *fptr3;
    char filename[100], line1[256], line2[256];
    char *r1, *r2;

    printf("Enter the first filename to open for reading: ");
    scanf("%s", filename);
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL) {
        printf("Cannot open file %s\n", filename);
        exit(0);
    }
    printf("Enter the second filename to open for reading: ");
    scanf("%s", filename);
    fptr2 = fopen(filename, "r");
    if (fptr2 == NULL) {
        printf("Cannot open file %s\n", filename);
        exit(0);
    }

    printf("Enter the filename to open for writing: ");
    scanf("%s", filename);
    fptr3 = fopen(filename, "w+");

    while (1) {
        r1 = fgets(line1, sizeof(line1), fptr1);
        r2 = fgets(line2, sizeof(line2), fptr2);
        if (r1 == NULL && r2 == NULL)
            break;
        if (r1 != NULL)
            fputs(line1, fptr3);
        if (r2 != NULL)
            fputs(line2, fptr3);
    }
    printf("Files merged into %s\n", filename);
    fclose(fptr1);
    fclose(fptr2);
    fclose(fptr3);
    return 0;
}
```

Input/Output:

```
Enter the first filename to open for reading: source.txt
Enter the second filename to open for reading: reversed.txt
Enter the filename to open for writing: merged.txt
Files merged into merged.txt
```

Contents of source.txt:

```
The w+ mode is designed for scenarios where you want to create a new file or completely replace the contents of an existing one.

    If the file exists: Its entire content is immediately truncated (deleted), and the file pointer is positioned at the beginning.
    If the file does not exist: A new, empty file is created.
    Pointer Position: The file pointer starts at the beginning of the file.
    Use Case: Ideal for writing new data from scratch and then potentially reading that data back within the same session.
```

Contents of reversed.txt:

```

 .noisses emas eht nihtiw kcab atad taht gnidaer yllaitnetop neht dna hctarcs morf atad wen gnitirw rof laedI :esaC esU
.elif eht fo gninnigeb eht ta strats retniop elif ehT :noitisoP retnioP
.detaerc si elif ytpme ,wen A :tsixe ton seod elif eht fI
.gninnigeb eht ta denoitisop si retniop elif eht dna ,)deteled( detacnurt yletaidemmi si tnetnoc eritne stI :stsixe elif eht fI
```

.eno gnitsixe na fo stnetnoc eht ecalper yletelpmoc ro elif wen a etaerc ot tnaw uoy erehw soiranecs
rof dengised si edom +w ehT

Contents of merged.txt:

The w+ mode is designed for scenarios where you want to create a new file or completely replace the contents of an existing one.

.noisses emas eht nihtiw kcab atad taht gnidaer yllaitnetop neht dna hctarcs morf atad wen gnitirw
rof laedI :esaC esU

If the file exists: Its entire content is immediately truncated (deleted), and the file pointer is positioned at the beginning.

.elif eht fo gninnigeb eht ta strats retniop elif ehT :noitisoP retnioP

If the file does not exist: A new, empty file is created.

.detaerc si elif ytpme ,wen A :tsixe ton seod elif eht fI

Pointer Position: The file pointer starts at the beginning of the file.

.gninnigeb eht ta denoitisop si retniop elif eht dna ,)deteled(detacnurt yletaidemmi si tnetnoc eritne
stI :stsixe elif eht fI

Use Case: Ideal for writing new data from scratch and then potentially reading that data back within the same session.

.eno gnitsixe na fo stnetnoc eht ecalper yletelpmoc ro elif wen a etaerc ot tnaw uoy erehw soiranecs
rof dengised si edom +w ehT