

Week1

Name : NEERAJ KUMAR

Regno. : 220905536

Roll no. : 57

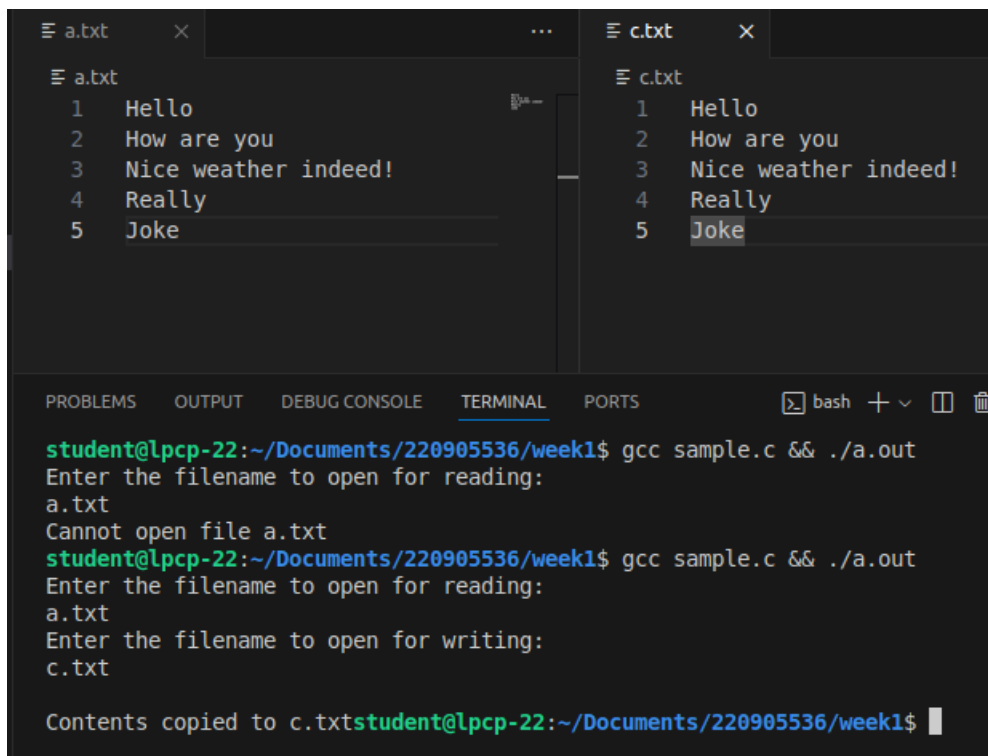
Title: BASIC FILE HANDLING OPERATIONS

Sample Exercise:

1. Write a C program to copy the contents of source file to destination file

```
#include <stdio.h>
#include <stdlib.h> // For exit()
int main()
{
    FILE *fptr1, *fptr2;
    char filename[100], c;
    printf("Enter the filename to open for reading: \n");
    scanf("%s", filename);
    fptr1 = fopen(filename, "r");
    // Open one file for reading
    if (fptr1 == NULL)
    {
        printf("Cannot open file %s \n", filename);
        exit(0);
    }
    printf("Enter the filename to open for writing: \n");
    scanf("%s", filename);
    fptr2 = fopen(filename, "w+"); // Open another file for writing
    c = fgetc(fptr1);
    // Read contents from file
    while (c != EOF)
    {
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }
    printf("\nContents copied to %s", filename);
    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

Output:



```
a.txt
1 Hello
2 How are you
3 Nice weather indeed!
4 Really
5 Joke

c.txt
1 Hello
2 How are you
3 Nice weather indeed!
4 Really
5 Joke

student@lpcp-22:~/Documents/220905536/week1$ gcc sample.c && ./a.out
Enter the filename to open for reading:
a.txt
Cannot open file a.txt
student@lpcp-22:~/Documents/220905536/week1$ gcc sample.c && ./a.out
Enter the filename to open for reading:
a.txt
Enter the filename to open for writing:
c.txt
Contents copied to c.txt
student@lpcp-22:~/Documents/220905536/week1$
```

Lab Exercise:

1) Write a C program to count the number of lines and characters in a file.

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

void countChar_Lines(char *filename)
{
    int charCount=0, lineCount=0;
    FILE *file1 = fopen(filename,"r");
    char c = fgetc(file1);
    while (c!=EOF)
    {
        charCount++;
        if (c=='\n')
        {
            lineCount++;
            charCount--;
        }
        c = fgetc(file1);
    }
    printf("\nNumber of charcters is: %d",charCount);
    printf("\nNumber of Lines in file is: %d\n",lineCount);
    fclose(file1);
}

int main()
{
    char filename[100];
    printf("Enter the filename to open for reading: \n");
    scanf("%s", filename);
    countChar_Lines(filename);
    return 0;
}
```

Output:

```
student@lpcp-22:~/Documents/220905536/week1$ gcc lab1_1.c && ./a.out
Enter the filename to open for reading:
a.txt

Number of characters is: 46
Number of Lines in file is: 4
student@lpcp-22:~/Documents/220905536/week1$
```

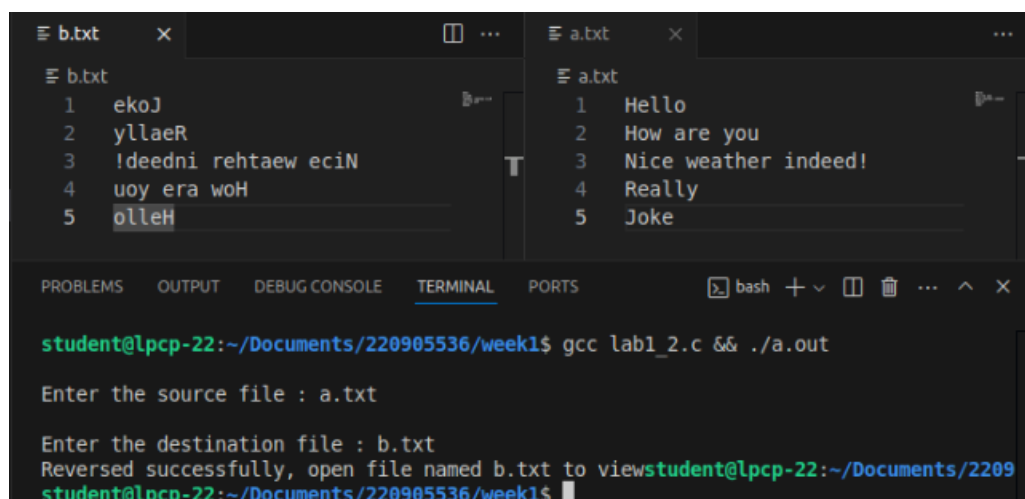
- 2) Write a C program to reverse the file contents and store in another file. Also display the size of file using file handling function.

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

```
void reverseFile(FILE *file1, FILE *file2)
{
    char c = fgetc(file1);
    if (c==EOF)
    {
        return;
    }
    reverseFile(file1,file2); //recursively invoking the function to reach the end of the file(EOF).
    fputc(c,file2);
}

int main()
{
    char srcFile[128], destFile[128];
    printf("\nEnter the source file : ");
    scanf("%s",srcFile);
    printf("\nEnter the destination file : ");
    scanf("%s", destFile);
    FILE *file1 = fopen(srcFile,"r");
    FILE *file2 = fopen(destFile,"w+");
    reverseFile(file1,file2);
    printf("Reversed successfully, open file named %s to view",destFile);
    fclose(file1);
    fclose(file2);
    return 0;
}
```

Output:



```
student@lpcp-22:~/Documents/220905536/week1$ gcc lab1_2.c && ./a.out

Enter the source file : a.txt

Enter the destination file : b.txt
Reversed successfully, open file named b.txt to view
student@lpcp-22:~/Documents/220905536/week1$
```

The screenshot shows a code editor with two files open: `b.txt` and `a.txt`. `b.txt` contains the reversed content of `a.txt`:

```
1 ekoJ
2 yllaeR
3 ldeedni rehtaew eciN
4 uoy era woH
5 olleH
```

`a.txt` contains the original content:

```
1 Hello
2 How are you
3 Nice weather indeed!
4 Really
5 Joke
```

Below the editor is a terminal window showing the execution of the C program. It prompts for the source file (`a.txt`) and the destination file (`b.txt`), then prints "Reversed successfully, open file named b.txt to view".

3. Write a C program that merges lines alternatively from 2 files and stores it in a resultant file.

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

```
void readLinesAtsameTime(char *srcfile1, char *srcfile2, char *destFile)
```

```
{
    char buffer[1024];
    FILE *file1 = fopen(srcfile1, "r");
    FILE *file2 = fopen(srcfile2, "r");
    FILE *file3 = fopen(destFile, "w+");

    // to read same lines from two files concurrently
    while (1)
    {
        if (fgets(buffer, sizeof(buffer), file1))
        {
            fputs(buffer, file3);
        }
        else
        {
            break;
        }
        if (fgets(buffer, sizeof(buffer), file2))
        {
            fputs(buffer, file3);
        }
        else
        {
            break;
        }
    }

    // if file2 is read but file1 not
    while (fgets(buffer, sizeof(buffer), file1))
    {
        fputs(buffer, file3);
    }

    // if file1 is read but file2 not
    while (fgets(buffer, sizeof(buffer), file2))
    {
        fputs(buffer, file3);
    }

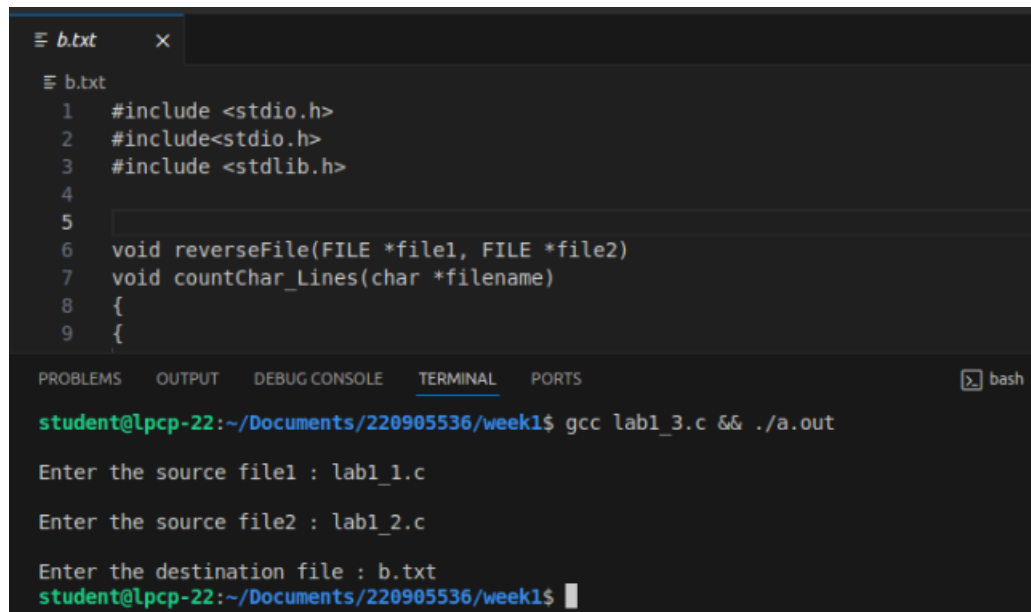
    fclose(file1);
    fclose(file2);
    fclose(file3);
}
```

```
int main()
```

```
{
    char file1[128], file2[128], destFile[128];
    printf("\nEnter the source file1 : ");
    scanf("%s", file1);
    printf("\nEnter the source file2 : ");
    scanf("%s", file2);
}
```

```
printf("\nEnter the destination file : ");
scanf("%s", destFile);
readLinesAtsameTime(file1, file2, destFile);
printf("Operation accomplished, open file named %s to view",destFile);
return 0;
}
```

Output: Here 2 times `#include<stdio.h>` is repeated, each from `lab1_1.c` and `lab1_2.c` files respectively.



The screenshot shows a code editor window with a file named `b.txt` open. The code in the editor is as follows:

```
1 #include <stdio.h>
2 #include<stdio.h>
3 #include <stdlib.h>
4
5
6 void reverseFile(FILE *file1, FILE *file2)
7 void countChar_Lines(char *filename)
8 {
9 {
```

Below the code editor is a terminal window. The terminal shows the command `gcc lab1_3.c && ./a.out` being executed. The output of the program is as follows:

```
student@lpcp-22:~/Documents/220905536/week1$ gcc lab1_3.c && ./a.out
Enter the source file1 : lab1_1.c
Enter the source file2 : lab1_2.c
Enter the destination file : b.txt
student@lpcp-22:~/Documents/220905536/week1$
```