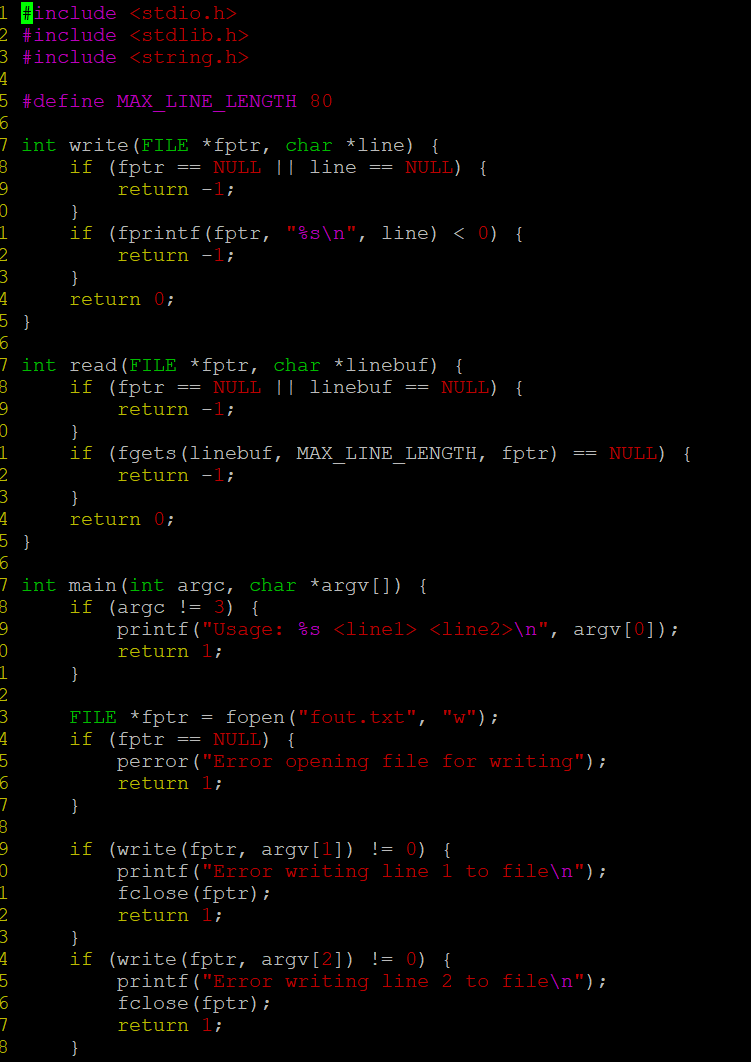
**FILE IO BASICS AND TEXT FILE HANDLING**

**1. Read 2 lines of text as single command line argument, validate the arguments, extract the lines, write to file “fout.txt”. Now open file read the content and display. Implement the functions**

**int write(FILE \*fptr, char \*line);**

**int read(FILE \*fptr, char \*linebuf);i**

**[Assume maximum line length as 80]**

****

**A computer screen with text on it

Description automatically generated**

**OUTPUT:**

**A black background with white text

Description automatically generated**

**2. Accept 3 file names as command line arguments. The first 2 are input files in which first file has to be created as an integer file and the second file has to created as a string file. Merge the contents of these 2 files into the 3rd file. It should be one integer from the first file followed by one line from the second file.**

**a. Display the merged file.**

**b. Add appropriate error handling.**

**c. Modularize the program and do it as multi file program.**

**d. Remove all memory leaks**

**e. Read "Integer file" using fscanf (Formatted I/O)**

**f. Read "Strings file" using fgets (Line I/O)**

**g. Write "Output file" using fprintf (Formatted I/O)**

**Example:**

**f1.txt**

**10**

**20**

**f2.txt:**

**hello**

**hi**

**fout.txt:**

**10hello**

**20hi**

**3. Copy the file “string\_process\_prg.c“ to your local directory. Consider a line length of 80 characters. Create “input.txt” file with appropriate data.**

**a) Fix the issues (warnings and errors in file).**

**b) Implement display()**

**c) Test the program for the expected output i.e to display file contents.**

**d) Free the allocated memory**

****

**OUTPUT:**

**A black screen with white text

Description automatically generated**