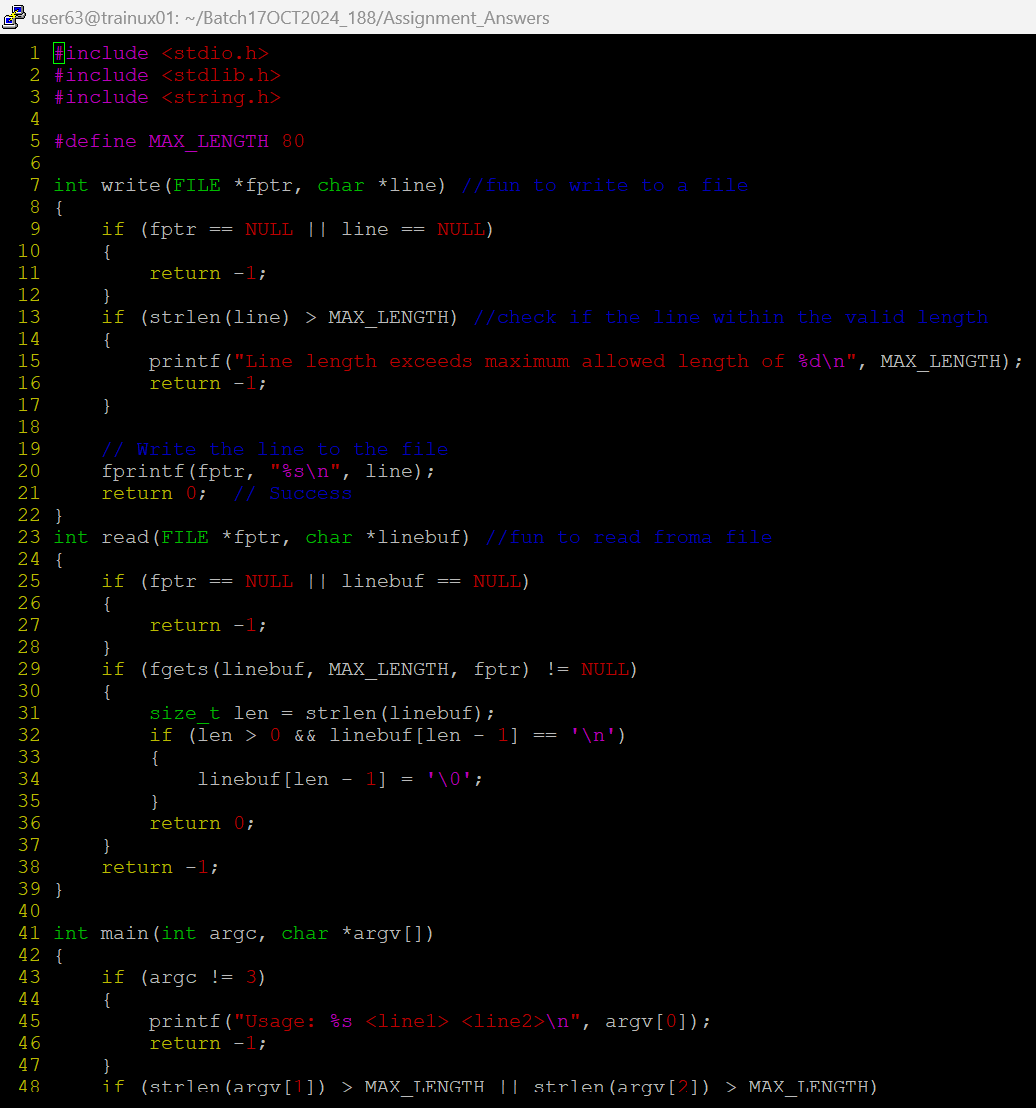
**File IO Basics and Text File Handling assignments**

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

A screenshot of a computer program

Description automatically generated[Assume maximum line length as 80]

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.

* 1. Display the merged file.
  2. Add appropriate error handling.
  3. Modularize the program and do it as multi file program.
  4. Remove all memory leaks
  5. Read "Integer file" using fscanf (Formatted I/O)
  6. Read "Strings file" using fgets (Line I/O)
  7. Write "Output file" using fprintf (Formatted I/O)

Example:

f1.txt

10

20

f2.txt:

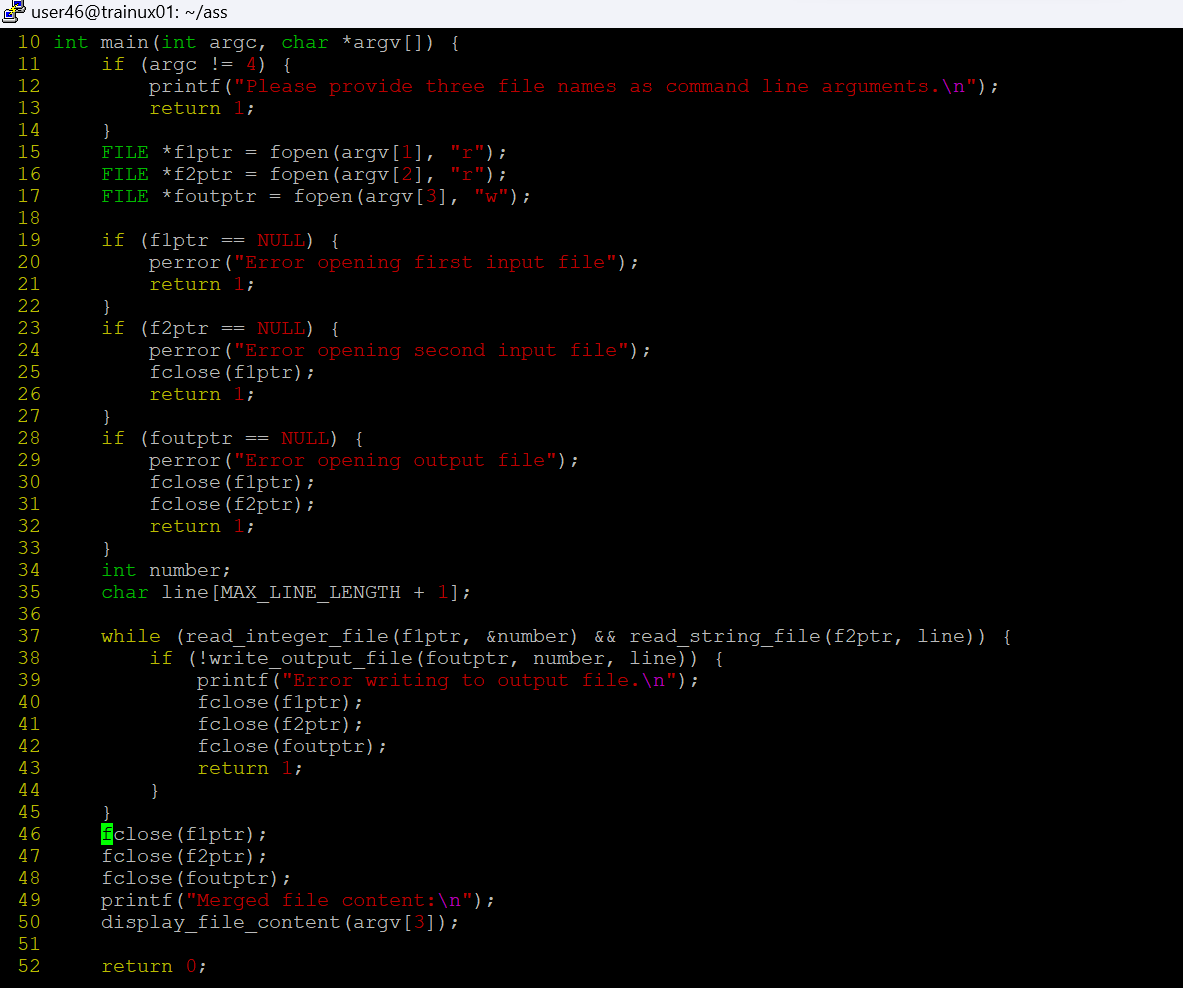
hello

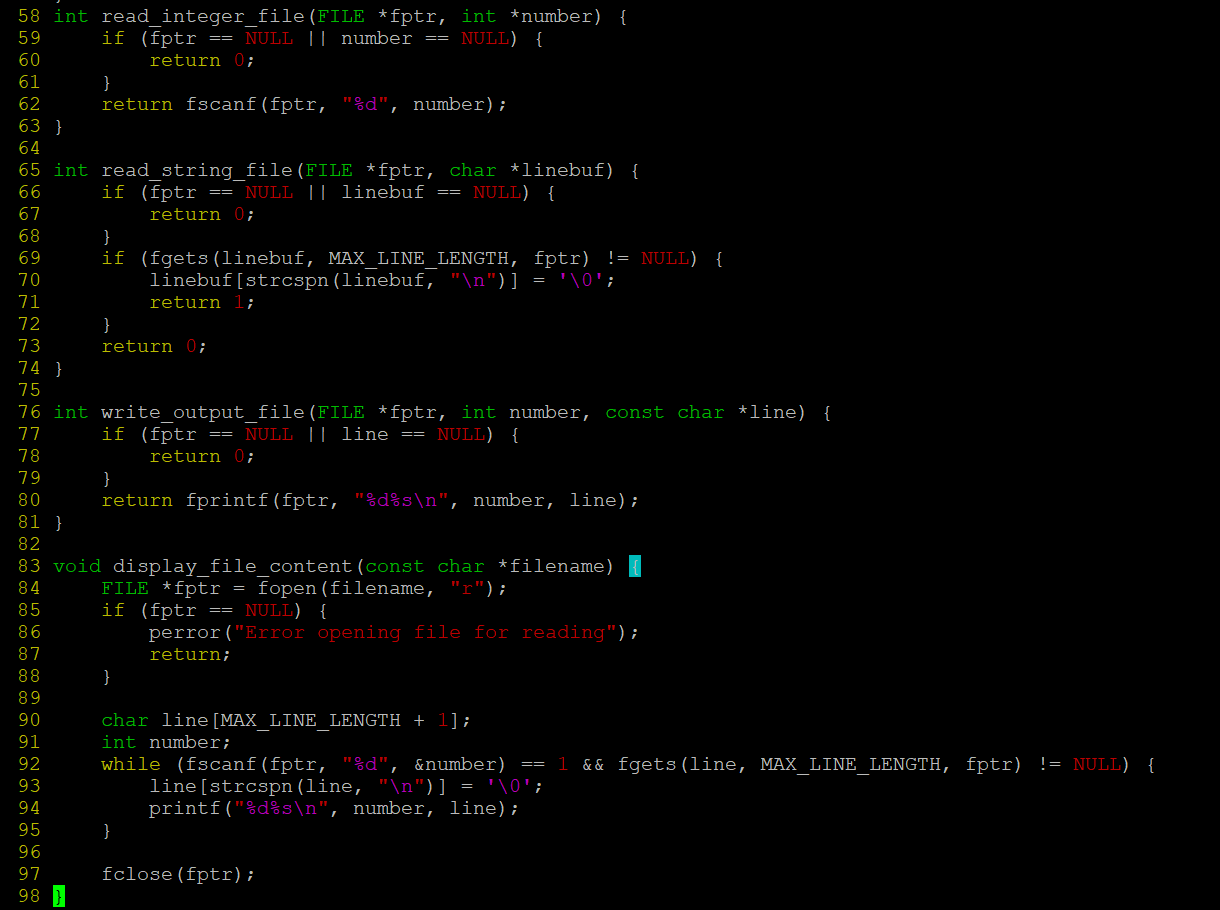
hi

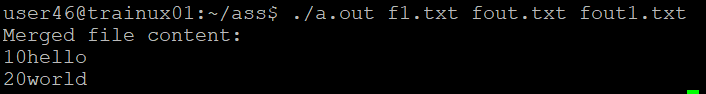
fout.txt:

10hello

20hi

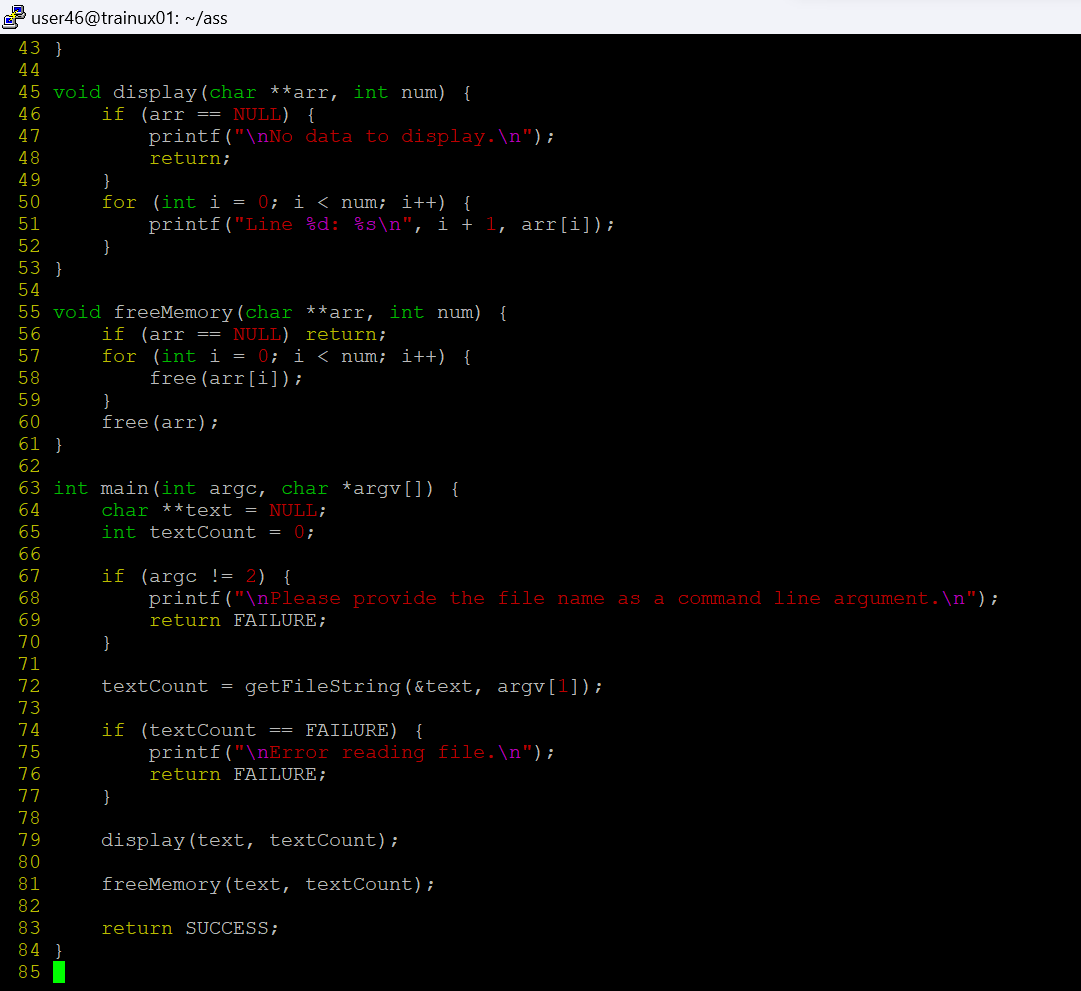


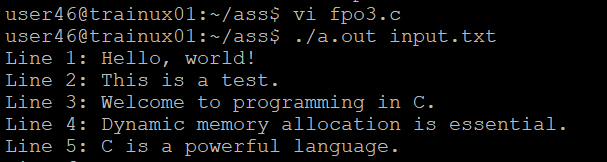




1. Copy the file “string\_process\_prg.c“ to your local directory. Consider a line length of 80 characters. Create “input.txt” file with appropriate data.
2. Fix the issues (warnings and errors in file).
3. Implement display()
4. Test the program for the expected output i.e to display file contents.
5. Free the allocated memory







1. 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.
2. Fix the issues (warnings and errors in file).
3. Implement display()
4. Test the program for the expected output i.e to display file contents.



1. Free the allocated memory

A screen shot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generated

5