Week 09, Lab 09	Weight: 1%	Due: End of your stream's week 9 lab session (via `sync`)

Pre-lab Preparation:

- **Week 1, 2, 3, 4, 5, 6, 7, 8 Lectures, Week 1, 2, 3, 4, 5, 6, 7, 8 Labs**
- **Week 9, Lecture 021**

Lab Activities:

Remember to **`sync`** to obtain the lab starting code.

Also, remember you can use the manual pages to learn more about library functionality covered in class, for example, try typing the following in Linux:

- **`man fopen`**

As you work through the lab exercises, if you encounter programming terminology that you do not understand:

- Refer to the Programming 1 lecture materials.
- Ask a Lab TA for clarification on what the term means.
- After attempting an activity, if you are stuck for an unreasonable amount of time, seek help from a Lab TA! Do not wait too long to seek help!

Exercise 1: Writing to a Text File

What is the purpose of the **`fopen`** *function?*

What is the purpose of the **`fprintf`** *function?*

What is the purpose of the **`fclose`** *function?*

Complete the following tasks, in order:

- Declare a `main` function in `lab09ex01.c` which returns zero.
- Call `fopen` to create a output file called `"ex1.txt"`
- Use `fprintf` to output the text `"Hello File World!"` to the `ex1.txt` file.
- Close the file stream with `fclose`.

Compile your program and run the resulting executable. Check the directory for the `ex1.txt` file using `ls`. If the file is present, run less on it using `less ex1.txt`.

Answer the following questions in the handout:

What does the `ex1.txt` *file contain?*

What is a file stream?

Exercise 2: Reading from a Text File

Explain the purpose of `fscanf`*?*

Complete the following tasks, in order:

- Declare a main function in `lab09ex02.c` which returns zero.
- Call `fopen` to open an input file called `"ex2.txt"`
- Use `fscanf` to read in the entire text file.
  - Read a byte at a time.
  - Print to the console the contexts of the file, one byte at a time.
  - At the start of each line printed to the console, print the line number of the file.
- Close the file stream with `fclose`.

Compile your program and run the resulting executable.

Example `ex2.txt` file on disk contents:

```
Hello...
Programming 1...
Students!
Welcome to File I/O!
```

Example `lab09ex02` program output:

```
1) Hello...
2) Programming 1...
3) Students!
4) Welcome to File I/O!
```

There are more `.txt` files to test your program with inside the `ex02` directory.

Have a lab TA review your completed exercises 1 and 2 for this lab session. See the end of this document for the review questions

Exercise 3: Command Line Arguments and File Output

How are command line arguments accessed inside a C program?

Write a program in **`lab09ex03.c`** which takes in command line arguments in the following format:

```
lab09ex03 3 Hello 5 Programming 2 Students! 10 :-)
```

Where the first argument after the name of the program represents the number of times to print the following argument. This pattern of "number of repetitions" and "text to repeat" can repeat.

Based upon the command line above, the following **`output.txt`** file should be written:

```
Hello, Hello, Hello
Programming, Programming, Programming, Programming, Programming
Students!, Students!
:-), :-), :-), :-), :-), :-), :-), :-), :-), :-)
```

Check the output with **`less output.txt`**.

Exercise 4: Binary Reading and Writing

What is the purpose of the `fread` *function?*

What is the purpose of the `fwrite` *function?*

Open the file `ex4in.bin` in binary read mode, the file contains four-byte integer values.

Read each integer value into your program, then square the value, print the result, and write then write it out to `ex4out.bin`.

Exercise 5: Byte Flipping

What does endian mean?

When a video game is developed on a Windows or Linux computer which runs an Intel x86 processor, the resulting game asset files are stored as "little endian". The Nintendo Wii, Microsoft Xbox 360 and Sony PlayStation 3 are all PowerPC-type processors, which is "big endian" by default. Hence all game asset files must be endian-swapped for use on these consoles.

Write a program in `lab09ex05.c` which reads in the file `pcdata.bin` and write out a endian swapped version of the file called `linuxdata.bin`.

For example: if **`0xABCD1234`** is written in the `pcdata.bin` file, the data must be written as **`0x3412CDAB`** in the corresponding position in the `linuxdata.bin` file.

Week 09, Lab 09 Submission:

Run the **sync** command to submit your completed lab work.

Shutdown your Raspberry Pi by pressing **ALT-CTRL-DEL**. Power-down and pack up your Raspberry Pi kit.

Marking Criteria:

Have you completed each of the following? Have you submitted your code from lab?

Marking Criteria:	Week 09 Lab 09 Weight 1%	Yes	No
Ex 1:	Code written correctly?		
	Correctly named file with right contents?		
	Questions answered correctly on handout?		
Ex 2:	Correct output from program?		
	Questions answered correctly on handout?		
Ex 3:	Questions answered correctly on handout?		
	Correct output from program?		
Ex 4:	Questions answered correctly on handout?		
	Correct output?		
	Correct relationship between input and output files? (use hexdump to check)		
Ex 5:	Questions answered correctly on handout?		
	Correct relationship between input and output files? (use hexdump to check)		

Next activity: Homework 7 and Final Week 9 Lecture