CS-392 Systems Programming Spring 2022

Midterm 1

Deadline: see Canvas

Write a C program that reads a few integer numbers, and print their big endian values. Your program should implement the following function, and call it in a main function:

```
1 void little2big(int* arr, int len);
```

where `arr` is the base address of the array where the integers are stored, and `len` the length of the array.

The integers you need to read are in a text file, one number each line. When you read in each number, it's stored in memory as little endian value. You need to convert each number into a big endian value, and write them into a file, one number each line. You don't need to store the numbers into an array. You can declare any type of function you like, but cannot change the prototype of `little2big()`.

Hint

Recall that an integer takes four bytes in memory. Suppose the four bytes of an integer from address `0x1000` to `0x1003` are `0x12`, `0x34`, `0x56`, and `0x78`. In little endian, the integer value will be `0x78563412`, while in big endian, it will be `0x12345678`.

You can cast a pointer back to `int*` and dereference to get the integer value starting at that pointer.

Requirements

Note the requirements are only part of the rubrics.

- You can assume there's no empty lines, or lines that contain non-numeric characters;
- The name of the input file is passed as a command-line argument `argv[1]`, while that of output file `argv[2]`;
- Your program shouldn't hardcode any variables;
- You shouldn't allocate space larger than necessary, and shouldn't have memory leak.

Sanity Check

To check your program is correct, you can use the following example. If the input is:

```
1
-1
0
23
9024
```

the output is

```
16777216
-1
0
385875968
107603584
```