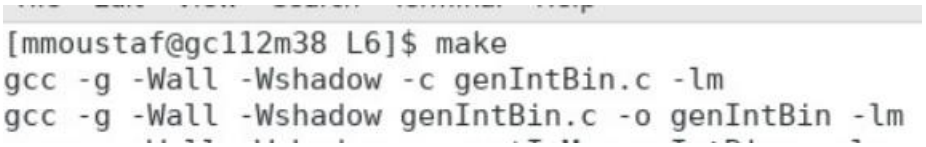
Lab 5

Exercise 1

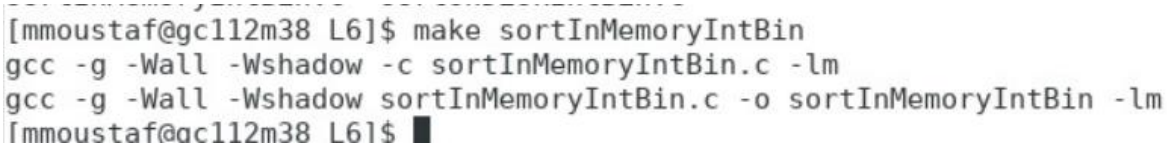
A screen shot of the make command output for a successful compile



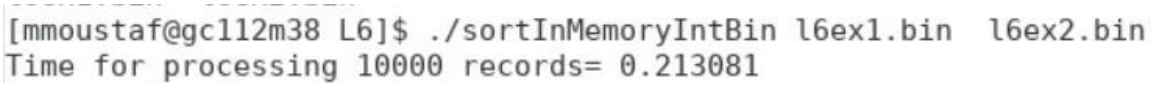
A screen shot of the program's successful run for 10,000 values

Exercise 2

A screen shot of the make command output for a successful compile



A screen shot of the program's successful run.



Exercise 3

A screen shot of the make command output for a successful compile.

```
[mmoustaf@gc112m38 L6]$ make sortOnDiskIntBin
gcc -g -Wall -Wshadow -c sortOnDiskIntBin.c -lm
gcc -g -Wall -Wshadow sortOnDiskIntBin.c -o sortOnDiskIntBin -lm
```

A screen shot of the program's successful run.

```
[mmoustaf@gc112m38 L6]$ cp l6ex1.bin l6ex3.bin
[mmoustaf@gc112m38 L6]$ ./sortOnDiskIntBin l6ex3.bin
Time for processing 10000 records= 196.543243
```

A brief set of thoughts about the relative speeds of running the two sorting programs

There is a huge difference between the running time of both algorithms. It took less than a second to run the first algorithm using memory, while it took more than 196 seconds to run the second algorithm on disk. I would always use the memory if that is the case.

Exercise 4

A screen shot of the make command output for a successful compile

```
[mmoustaf@gc112m38 L6]$ make reportIntBin
gcc -g -Wall -Wshadow -c reportIntBin.c -lm
gcc -g -Wall -Wshadow reportIntBin.c -o reportIntBin -lm
```