1. Try the “make” command. What happens? Use “ls” to inspect your directory.

```
[mmoustaf@remotelabm18 Lab1]$ make
gcc -o hellomake hellomake.c hellofunc.c -I. -Wall
[mmoustaf@remotelabm18 Lab1]$ ls
hellofunc.c  hellomake  hellomake.c  hellomake.h  makefile
```

The gcc command is typed out and a new executable file called “makefile” is created

2.

a. what is the output? Why is this happening?

NO ERROR

b.

```
[mmoustaf@remotelabm18 Lab1]$ make
gcc -Wall -I.   -c -o hellomake.o hellomake.c
gcc -Wall -I.   -c -o hellofunc.o hellofunc.c
gcc -o hellomake hellomake.o hellofunc.o -Wall -I.
[mmoustaf@remotelabm18 Lab1]$ ls
hellofunc.c  hellomake    hellomake.h  makefile
hellofunc.o  hellomake.c  hellomake.o
```

We have 2 new gcc commands (the first 2). This is to compile the hellomake.c and hellofunc.c because we need them compiled (because we need their object files) to run the final command.

3.

```
[mmoustaf@remotelabm18 Lab1]$ make
make: `hellomake' is up to date.
```

it says hellomake is up to date

4.

```
[mmoustaf@remotelabm18 Lab1]$ make
gcc -c hellomake.c -Wall -I.
gcc -c hellofunc.c -Wall -I.
gcc -o hellomake hellomake.o hellofunc.o -Wall -I.
```

```
[mmoustaf@remotelabm18 Lab1]$ make run
./hellomake
Hello makefiles! 63
```