Q1

a)

```
/* Reads in a data file consisting of 2 columns of 100 integer numbers
* Author - 2467273
* Assumes columns are separated by a comma.
*/
#include <stdio.h>
#include <string.h>
```

b)

```
int main(void){
        FP * fp = fopen("data.dat", "r");
        fclose(fp);
}
```

c)

```
int main(void){
        FP * fp = fopen("data.dat", "r");
        int columns = 2
        int rows = 100;
        int data[rows][columns];
        fclose(fp);
}
```

d) (all)

```
/* Reads in a data file consisting of 2 columns of 100 integer numbers
* Author - 2467273
* Assumes columns are separated by a comma.
*/
```

```
#include <stdio.h>
#include <string.h>


int main(void){
	FP * fp = fopen("data.dat", "r");
	int columns = 2
	int rows = 100;
	int data[rows][columns];

	int i = 0;
	for (i; i < rows; i++){
		char line[100];
		fgets(line, sizeof(line), stdin);
		sscanf(line, "%d,%d", &data[i][0], &data[i][1]);
	}
	fclose(fp);
	return 0;
}
```

e) When a user runs **make**, all necessary object files are created from the .c files present in the directory (access.o, util.o, main.o), and a **stats** executable is created from the compilation of all the object files. If run with **test** argument, the executable is run with the -h argument and the website given in the Makefile.

f) $@ is the target, i.e., $(EXE) (stats). $^ represents all of the prerequisites for the target, i.e., $(OBJS) (access.o, util.o, main.o).

Full line would be:

gcc access.o util.o main.o -o stats

g)

i. If **db.c** is present in the directory, add “db.o” to the list of OBJS (after main.o and a space)

ii. Add “clean” in line 12 (after “test” and a space) and on line 15:

clean:

	rm -f *.o $(EXE)