

Question 1:

```
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ nano file.txt
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ cat file.txt
this is a test.
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ln file.txt har
link.txt
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ls -l
total 8
-rw-rw-r-- 2 tallulah tallulah 16 Feb 13 17:50 file.txt
-rw-rw-r-- 2 tallulah tallulah 16 Feb 13 17:50 hardlink.txt
-rw-rw-r-- 1 tallulah tallulah  0 Feb 13 17:46 q1.c
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ln -s file.txt
oftlink.txt
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ls -l
total 8
-rw-rw-r-- 2 tallulah tallulah 16 Feb 13 17:50 file.txt
-rw-rw-r-- 2 tallulah tallulah 16 Feb 13 17:50 hardlink.txt
-rw-rw-r-- 1 tallulah tallulah  0 Feb 13 17:46 q1.c
lrwxrwxrwx 1 tallulah tallulah  8 Feb 13 17:50 softlink.txt -> file.txt
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$
```

First I created a hardlink to file.txt, which created its own file as you can see in the first ls -l command. Then I created a softlink to file.txt, which basically created a pointer to the file, as you can see in the second ls -l command.

Question 2:

```
4 int main(int argc, char *argv[]){
5     pthread_t threads[NUM_THREADS];
6     int rc;
7     rc = pthread_create(&threads[0], NULL, calc_max, NULL);
8     if (rc){
9         printf("Error, unable to create thread.\n");
0         exit(-1);
1     }
2     rc = pthread_create(&threads[1], NULL, calc_min, NULL);
3     if(rc){
4         printf("Error, unable to create thread.\n");
5         exit(-1);
6     }
7
8     for(int i=0; i<NUM_THREADS;i++){
9         rc = pthread_join(threads[i],NULL);
0         if (rc){
1             printf("Error, unable to join thread.\n");
2             exit(-1);
3         }
4
5         printf("The maximum value is %d\n", max);
6         printf("The minimum value is %d\n", min);
7
8         pthread_exit(NULL);
```

```
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ gcc q1.c -o q1
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ./q1
The maximum value is 95
The minimum value is 2
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$
```

For this one I passed in an array of integers and had two separate threads work through the array to determine the maximum and minimum values.

Question 3:

```
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ gcc -g q3.c -o q3
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ ./q3
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ cat lab4Output.txt
cat: lab4Output.txt: No such file or directory
tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$ cat outputLab4.txt
This is a test for opening, writing, and closing a file!tallulah@tallulah-VirtualBox:~/tmp/OS/CWU/LabAssignments/Lab4$
```

For this question I created a C file that opens a file called outputLab4 and writes to it the phrase "This is a test for opening, writing, and closing a file!". To do this I used the open() system call to open the file. And I used the write() system call to write to the file. Then I used close() to close the file.

Question 4:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <pthread.h>
5 #define CORE 4
6 #define MAX 4
7 pthread_t thread[CORE*3];
8 int mat_A[MAX][MAX], mat_B[MAX][MAX], sum[MAX][MAX], sub[MAX][MAX],
   product[MAX][MAX];
9
10 void* addition(void* arg){
11     int i,j;
12     int core = (int)arg;
13
14     for(i=core*MAX/4; i< (core+1)* MAX/4;i++){
15         for(j=0; j< MAX; j++){
16             sum[i][j] = mat_A[i][j] + mat_B[i][j];
17         }
18     }
19 }
20 void* subtraction(void* arg){
21     int i,j;
22     int core = (int)arg;
23
24     for(i=core*MAX / 4; i< (core + 1)* MAX / 4; i++){
```

Matrix A:

```
3736
9203
0217
2279
```

Matrix B:

```
6552
1796
6689
0352
```

Sum of Matrix A and B:

```
9  12  8  8
10 9  9  9
6  8  9  16
2  5  12 11
```

Difference of Matrix A and B:

| | | | |
|----|----|----|----|
| -3 | 2 | -2 | 4 |
| 8 | 5 | -9 | -3 |
| -6 | -4 | -7 | -2 |
| 2 | -1 | 2 | 7 |

Product of Matrix A and B:

| | | | |
|----|----|----|----|
| 18 | 35 | 15 | 12 |
| 9 | 14 | 0 | 18 |
| 0 | 12 | 8 | 63 |
| 0 | 6 | 35 | 18 |

To do this question I first created 2 matrices in the main method, populating the two with values under 10 for simplicity. Then I created threads equal to the core size and computed the matrix rows. The three different computations were done by having each thread compute $\frac{1}{4}$ of the matrix operation performed. After joining threads post computation, I displayed the three new matrices created for addition, subtraction, and multiplication.