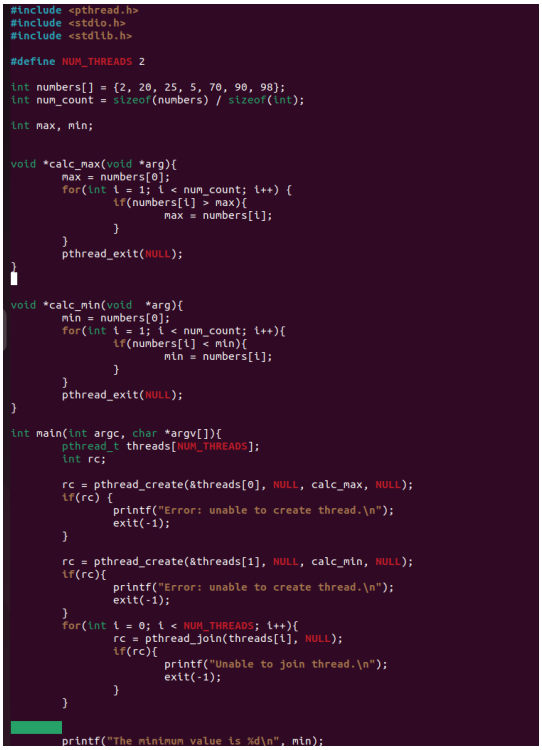
1.

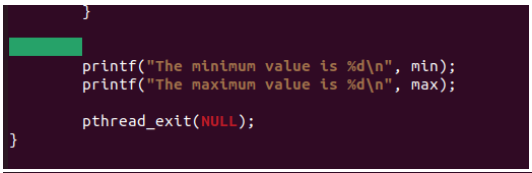
```
paulco@paulco-VirtualBox:~/lab4$ ln -s file.txt softlink.txt
paulco@paulco-VirtualBox:~/lab4$ ls -l
total 8
-rw-rw-r-- 2 paulco paulco 16 Feb 22 17:09 file.txt
-rw-rw-r-- 2 paulco paulco 16 Feb 22 17:09 hardlink.txt
-rw-rw-r-- 1 paulco paulco  0 Feb 22 16:46 matrix.c
lrwxrwxrwx 1 paulco paulco  8 Feb 22 17:15 softlink.txt -> file.txt
paulco@paulco-VirtualBox:~/lab4$
```

```
paulco@paulco-VirtualBox:~/lab4$ nano file.txt
paulco@paulco-VirtualBox:~/lab4$ touch file.txt
paulco@paulco-VirtualBox:~/lab4$ ls
file.txt  matrix.c
paulco@paulco-VirtualBox:~/lab4$ nano file.txt
paulco@paulco-VirtualBox:~/lab4$ cat file.txt
THis is a test!
paulco@paulco-VirtualBox:~/lab4$ ln file.txt hardlink.txt
paulco@paulco-VirtualBox:~/lab4$ ls -l
total 8
-rw-rw-r-- 2 paulco paulco 16 Feb 22 17:09 file.txt
-rw-rw-r-- 2 paulco paulco 16 Feb 22 17:09 hardlink.txt
-rw-rw-r-- 1 paulco paulco  0 Feb 22 16:46 matrix.c
paulco@paulco-VirtualBox:~/lab4$
```

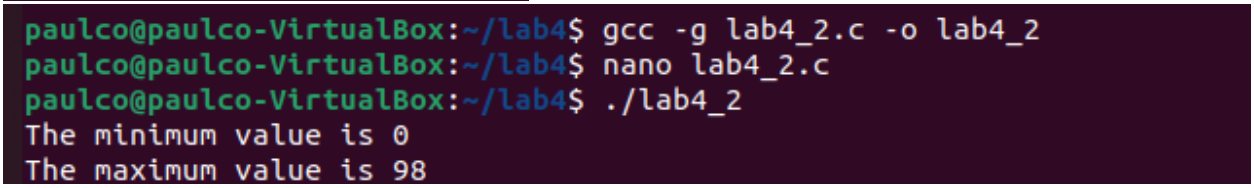
2.



```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define NUM_THREADS 2

int numbers[] = {2, 20, 25, 5, 70, 90, 98};
int num_count = sizeof(numbers) / sizeof(int);

int max, min;


void *calc_max(void *arg){
        max = numbers[0];
        for(int i = 1; i < num_count; i++) {
                if(numbers[i] > max){
                        max = numbers[i];
                }
        }
        pthread_exit(NULL);
}

void *calc_min(void  *arg){
        min = numbers[0];
        for(int i = 1; i < num_count; i++){
                if(numbers[i] < min){
                        min = numbers[i];
                }
        }
        pthread_exit(NULL);
}

int main(int argc, char *argv[]){
        pthread_t threads[NUM_THREADS];
        int rc;

        rc = pthread_create(&threads[0], NULL, calc_max, NULL);
        if(rc) {
                printf("Error: unable to create thread.\n");
                exit(-1);
        }

        rc = pthread_create(&threads[1], NULL, calc_min, NULL);
        if(rc){
                printf("Error: unable to create thread.\n");
                exit(-1);
        }
        for(int i = 0; i < NUM_THREADS; i++){
                rc = pthread_join(threads[i], NULL);
                if(rc){
                        printf("Unable to join thread.\n");
                        exit(-1);
                }
        }

        printf("The minimum value is %d\n", min);
```



```c
        }

        printf("The minimum value is %d\n", min);
        printf("The maximum value is %d\n", max);

        pthread_exit(NULL);
}
```



```
paulco@paulco-VirtualBox:~/lab4$ gcc -g lab4_2.c -o lab4_2
paulco@paulco-VirtualBox:~/lab4$ nano lab4_2.c
paulco@paulco-VirtualBox:~/lab4$ ./lab4_2
The minimum value is 0
The maximum value is 98
```

The main() function creates an array of integers, initializes some variables, and creates the three threads. The calc_max() and calc_min() functions calculate the maximum and minimum values of the array, respectively, by iterating over all the elements and comparing each element to the current maximum or minimum value.

3.

```
paulco@paulco-VirtualBox:~/lab4$ nano writeToFile.c
paulco@paulco-VirtualBox:~/lab4$ gcc -g writeToFile.c -o writeToFile
paulco@paulco-VirtualBox:~/lab4$ ./writeToFile
paulco@paulco-VirtualBox:~/lab4$ cat outputLab4.txt
This is a Test for opening, writing, and closing a file!paulco@paulco-VirtualBox
:~/lab4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main() {
        int fd;
        char buf[100] = "This is a Test for opening, writing, and closing a file!";
        ssize_t n;

        fd = open("outputLab4.txt", O_WRONLY | O_CREAT, 0644);
        if(fd == -1){
                perror("open");
                exit(EXIT_FAILURE);
        }

        n = write(fd, buf, sizeof(buf));
        if(n == -1){
                perror("write");
                exit(EXIT_FAILURE);
        }

        if(close(fd) == -1){
                perror("close");
                exit(EXIT_FAILURE);
        }

        return 0;
}
```

fd = open("outputchange.txt", O_WRONLY | O_CREAT, 0644);

^ This calls the open function to open the file named "outputchange.txt" in write-only mode (O_WRONLY). If the file doesn't exist, it will be created (O_CREAT).