4.22 Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value.
For example, suppose your program is passed the integers 90 81 78 95 79 72 85
The program will report

The average value is 82
The minimum value is 72
The maximum value is 95

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

#define ARRAY_SIZE 7
#define NUM_THREADS 3
int arr[ARRAY_SIZE];
int avg;
int min,max;
void *avgarr(void *arg)
{
int sum=0;
for(int i=0;i<ARRAY_SIZE;i++)
{
sum += arr[i];
}
avg =(int)sum/ARRAY_SIZE;
pthread_exit(0);
}
void *minarr(void *arg)
{
min=arr[0];
for(int i=1;i<ARRAY_SIZE;i++)
{
if(arr[i] < min)
{
min = arr[i];
}
}
pthread_exit(0);
}
void *maxarr(void *arg)
{
max=arr[0];
for(int i=1;i<ARRAY_SIZE;i++)
{
if(arr[i] > max)
{
max = arr[i];
}
}
pthread_exit(0);
}
```

```c
int main(int argc, char *argv[])
{
if(argc!=8)
{
printf("Usage: %s <arrayelements>", argv[0]);
return 1;
}
for(int i=0;i<7;i++)
{
arr[i]=atoi(argv[i+1]);
}
pthread_t workers[3];
pthread_create(&workers[0], NULL, avgarr, NULL);
pthread_create(&workers[1], NULL, minarr, NULL);
pthread_create(&workers[2], NULL, maxarr, NULL);
    for (int i = 0; i < NUM_THREADS; i++)
    {
        pthread_join(workers[i], NULL);
    }

    printf("Average: %d\n", avg);
    printf("Min: %d\n", min);
    printf("Max: %d\n", max);
return 0;
}

```

```
student@VW:~$ gedit multithread.c
student@VW:~$ gcc multithread.c -o out
student@VW:~$ ./out 90 81 78 95 79 72 85
Average: 82
Min: 72
Max: 95
```

4.23 Write a multithreaded program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include<unistd.h>
int isPrime(int num)
{
if(num<=1)
{
return 0;
}
for(int i=2;i*i<=num;i++)
{
if(num%i ==0)
{
return 0;
}
return 1;
}
}
void *prime_printer(void *arg) {
    int limit = *((int *)arg);
    free(arg);
    printf("Primes up to %d: ", limit);
    for (int i = 2; i <= limit; i++) {
        if (isPrime(i)) {
            printf("%d ", i);
        }
    }
    printf("\n");
    return NULL;
}
int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <number>\n", argv[0]);
        return 1;
    }

    int limit = atoi(argv[1]);
    if (limit < 2) {
        printf("No primes less than or equal to %d\n", limit);
        return 0;
    }
   int *limit_ptr = malloc(sizeof(int));
    *limit_ptr = limit;
    pthread_t thread;
    if (pthread_create(&thread, NULL, prime_printer, limit_ptr) != 0) {
        perror("Failed to create thread");
        free(limit_ptr);
        return 1;
    }
    pthread_join(thread, NULL);
    return 0;
}
```

```
k200281kainat@k200281kainat-VirtualBox:~$ gcc primes.c -o out -pthread
k200281kainat@k200281kainat-VirtualBox:~$ ./out 25
Primes up to 25: 2 3 5 7 9 11 13 15 17 19 21 23 25
```