Program

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

double mean = 0;
double median = 0;
double stddev = 0;

int *numbers = NULL;
int num_count = 0;

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;

void *calculate_mean(void *arg){
        double sum = 0;
        for(int i = 0;i<num_count;i++){
                sum += numbers[i];
        }
        pthread_mutex_lock(&mutex);
        mean = sum/num_count;
        pthread_mutex_unlock(&mutex);
        pthread_exit(NULL);
}

void *calculate_median(void *arg){
        for(int i = 0;i<num_count-1;i++){
                for(int j = i+1;j<num_count;j++)
                {
                        if(numbers[i]>numbers[j]){
                                int temp = numbers[i];
                                numbers[i] = numbers[j];
                                numbers[j] = temp;
                        }
                }
        }
        pthread_mutex_lock(&mutex);
        if(num_count%2 == 0){
                median = (numbers[num_count/2-1] + numbers[num_count/2])/2;
        }
        else{
                median = numbers[num_count/2];
        }
        pthread_mutex_unlock(&mutex);
```

```
        pthread_exit(NULL);
}

void *calculate_stddev(void *arg){
        double sum = 0;
        pthread_mutex_lock(&mutex);
        for(int i = 0;i<num_count;i++){
                sum += pow(numbers[i]-mean,2);
        }
        stddev = sqrt(sum/num_count);
        pthread_mutex_unlock(&mutex);
        pthread_exit(NULL);
}

int main(int argc,char *argv[]){
        if(argc<2){
                printf("Not enough arguments\n");
                return EXIT_FAILURE;
        }
        num_count = argc-1;
        numbers = malloc(num_count*sizeof(int));
        if(!numbers){
                perror("malloc");
                return EXIT_FAILURE;
        }
        for(int i = 0;i<num_count;i++){
                numbers[i] = atoi(argv[i+1]);
        }
        pthread_t mean_thread,median_thread,stddev_thread;
        pthread_create(&mean_thread,NULL,calculate_mean,NULL);
        pthread_create(&median_thread,NULL,calculate_median,NULL);
        pthread_create(&stddev_thread,NULL,calculate_stddev,NULL);
        pthread_join(mean_thread,NULL);
        pthread_join(median_thread,NULL);
        pthread_join(stddev_thread,NULL);
        printf("MEAN:%.2f\n",mean);
        printf("MEDIAN:%.2f\n",median);
        printf("STD DEV:%.2f\n",stddev);
        free(numbers);
        pthread_mutex_destroy(&mutex);
        return EXIT_SUCCESS;
}
```

1 Sample run of the program

```
s23a40@Server-2:~/blab$ gcc exp15.c -pthread -lm
s23a40@Server-2:~/blab$ ./a.out 4 5 6
MEAN:5.00
MEDIAN:5.00
STD DEV:0.82
s23a40@Server-2:~/blab$ ./a.out 4 53 62
MEAN:39.67
MEDIAN:53.00
STD DEV:25.49
```