

# Lab 3 Report

**Name: Insiyah Ujjainwala**

**Email: [ujjainwi@mcmaster.ca](mailto:ujjainwi@mcmaster.ca)**

**Student ID: 400357483**

**Date: 1/11/2023**

## Stat.c

- This C program performs statistical calculations on a list of numbers. It uses three separate threads to compute the average, minimum, and maximum values of the input data.
- The **avg\_thread()** function calculates the average of the numbers. It prompts the user to enter the size of the list and then the individual numbers. After collecting the data, it calculates the average and prints the result.
- The **min\_thread()** function determines the minimum value from the input list. It iterates through the array and keeps track of the smallest value, which it prints.
- The **max\_thread()** function finds the maximum value in a similar way to min\_thread() but tracks the largest value instead.

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

int size[25] , n , i;

void *avg_thread()
{
    int sum = 0;
    float average;

    printf("Enter the size of your input list (between 1 to 25) = ");
    scanf("%d",&n);

    for(i=0; i<n; i++)
    {
        printf("Enter item %d:", i+1);
        scanf("%d",&size[i]);
    }
    for(i=0; i<n; i++)
    {
        sum=sum+size[i];
    }

    average=sum/n;

    printf("The average value is %f",average);
}

void *min_thread()
{

```

```

    int i;

    int temp = size[0];
    for(i=1; i<n; i++)
    {
        if(temp > size[i])
        {
            temp = size[i];
        }
    }
    printf("\nThe Minimum value is %d",temp);
}

void *max_thread()
{
    int i;

    int temp = size[0];
    for(i=1;i<n;i++)
    {
        if(temp < size[i])
        {
            temp = size[i];
        }
    }
    printf("\nThe Maximum value is %d\n",temp);
}

int main()
{
    pthread_t thread1;
    pthread_t thread2;
    pthread_t thread3;

    int i;
    int n;

    n = pthread_create(&thread1,NULL,&avg_thread,NULL);
    pthread_join(thread1,NULL);

    n = pthread_create(&thread2,NULL,&min_thread,NULL);
    pthread_join(thread2,NULL);

    n=pthread_create(&thread3,NULL,&max_thread,NULL);
    pthread_join(thread3,NULL);}

```

## Output

```
• insiyah@cs3sh3:~/Desktop$ ./stat
Enter the size of your input list (between 1 to 25) = 5
Enter item 1:12
Enter item 2:89
Enter item 3:45
Enter item 4:93
Enter item 5:105
The average value is 68.000000
The Minimum value is 12
The Maximum value is 105
```

## Prime.c

- This C program checks and prints prime numbers up to a given limit.
- The `is_prime_num` function determines if a given number is prime. It first checks for some basic cases (less than or equal to 1, equal to 2, or even), and then it iterates to check for divisibility up to the square root of the number, incrementing by 2 for efficiency.
- The `prime_nums` function is responsible for printing prime numbers up to a specified limit. It takes a limit as an argument and uses a loop to check and print prime numbers from 2 to the given limit.

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

int num;
// Function to check if a number is prime
int is_prime_num(num) {
    if (num <= 1) {
        return 0;
    }
    if (num == 2) {
        return 1;
    }
    if (num % 2 == 0) {
        return 0;
    }
    for (int i = 3; i * i <= num; i += 2) {
```

```

        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}

// Function to print prime numbers up to a given limit
void* prime_nums(void* limit) {
    int num;
    int n = *((int*)limit);
    for (num = 2; num <= n; num++) {
        if (is_prime_num(num)) {
            printf("%d\n", num);
        }
    }
    pthread_exit(NULL);
}

int main() {
    int number;
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    if (n < 2) {
        printf("Invalid number, try again\n");
        return 1;
    }

    pthread_t prime_thread;

    pthread_create(&prime_thread, NULL, prime_nums, &n);
    pthread_join(prime_thread, NULL);

    return 0;
}

```

## Output

```
● insiyah@cs3sh3:~/Desktop$ ./prime
```

```
Enter a number: 100
```

```
2
```

```
3
```

```
5
```

```
7
```

```
11
```

```
13
```

```
17
```

```
19
```

```
23
```

```
29
```

```
31
```

```
37
```

```
41
```

```
43
```

```
47
```

```
53
```

```
59
```

```
61
```

```
67
```

```
71
```

```
73
```

```
79
```

```
83
```

```
89
```

```
97
```