Lab 4 Task 1 + Task 2

Name: Muhammad Sherjeel Akhtar

Roll No: 20p-0101

Subject: Operating Systems Lab

Submitted To Respected Sir: Muhammad Ahsan

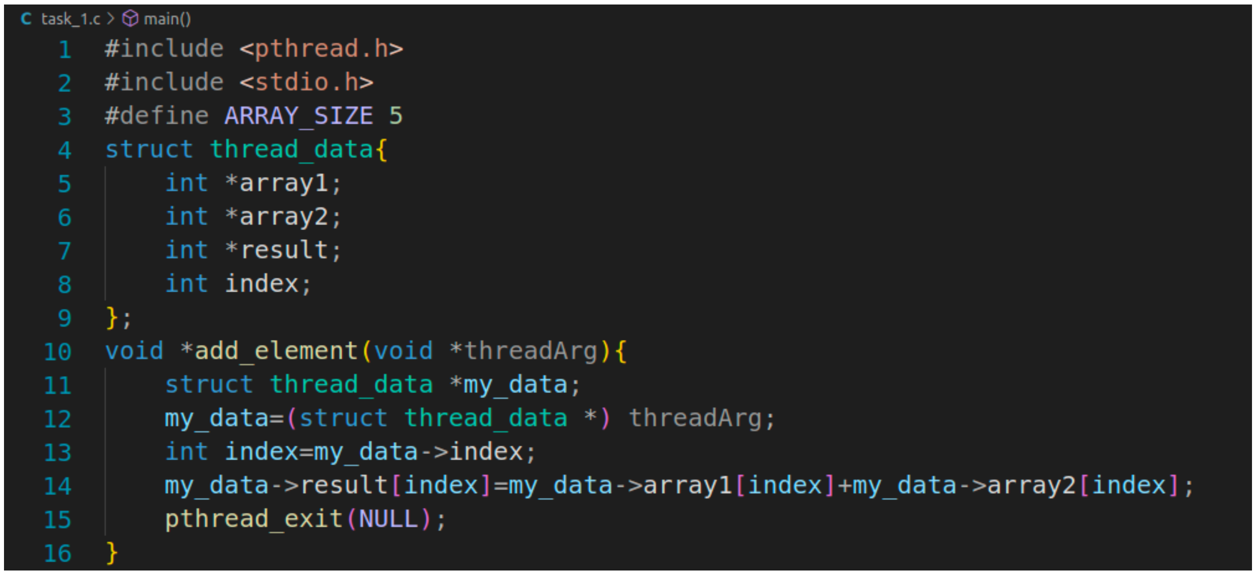
Section: BCS-4A

####786####

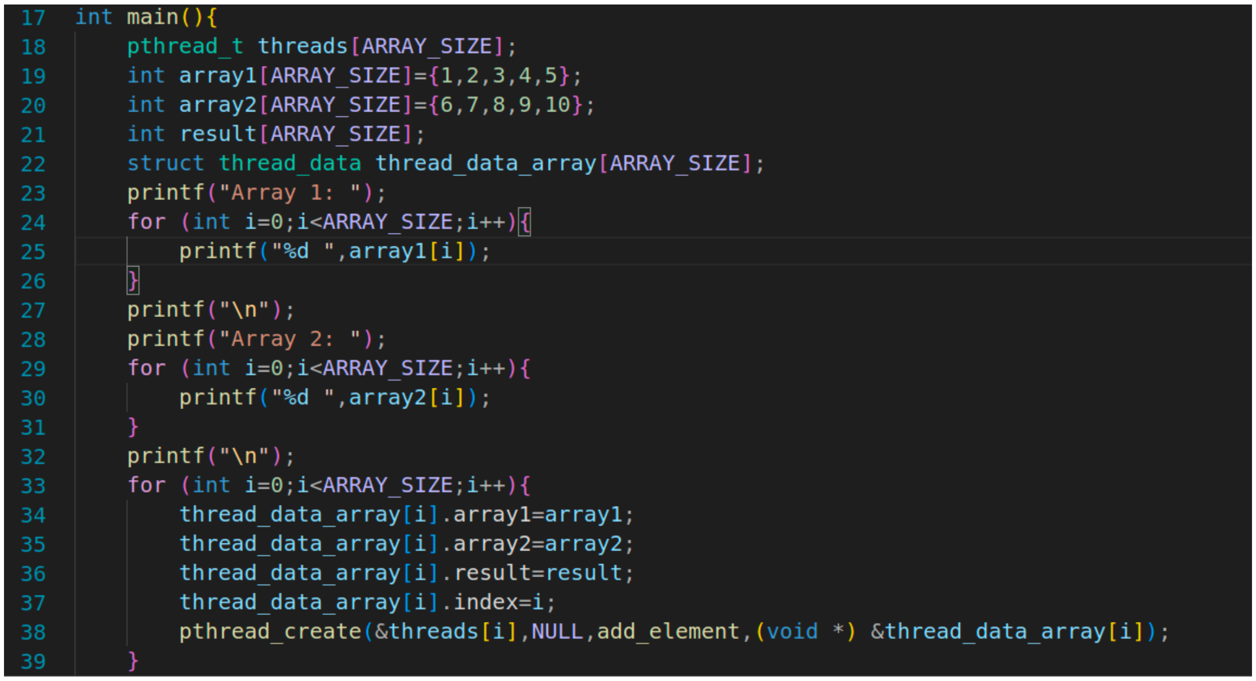
Task 1:

```
#include <pthread.h>
#include <stdio.h>
#define ARRAY_SIZE 5
struct thread_data{
    int *array1;
    int *array2;
    int *result;
    int index;
};
void *add_element(void *threadArg){
    struct thread_data *my_data;
    my_data=(struct thread_data *) threadArg;
    int index=my_data->index;
    my_data->result[index]=my_data->array1[index]+my_data->array2[index];
    pthread_exit(NULL);
}
int main() {
    pthread_t threads[ARRAY_SIZE];
    int array1[ARRAY_SIZE]={1,2,3,4,5};
    int array2[ARRAY_SIZE]={6,7,8,9,10};
    int result[ARRAY_SIZE];
    struct thread_data thread_data_array[ARRAY_SIZE];
    printf("Array 1: ");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",array1[i]);
    }
    printf("\n");
    printf("Array 2: ");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",array2[i]);
    }
    printf("\n");
    for (int i=0;i<ARRAY_SIZE;i++){
        thread_data_array[i].array1=array1;
        thread_data_array[i].array2=array2;
        thread_data_array[i].result=result;
        thread_data_array[i].index=i;
        pthread_create(&threads[i],NULL,add_element,(void *) &thread_data_array[i]);
    }
    for (int i=0;i<ARRAY_SIZE;i++){
        pthread_join(threads[i],NULL);
    }
    printf("Result:");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",result[i]);
    }
    printf("\n");
    printf("Exiting\n");
    return 0;
}
```

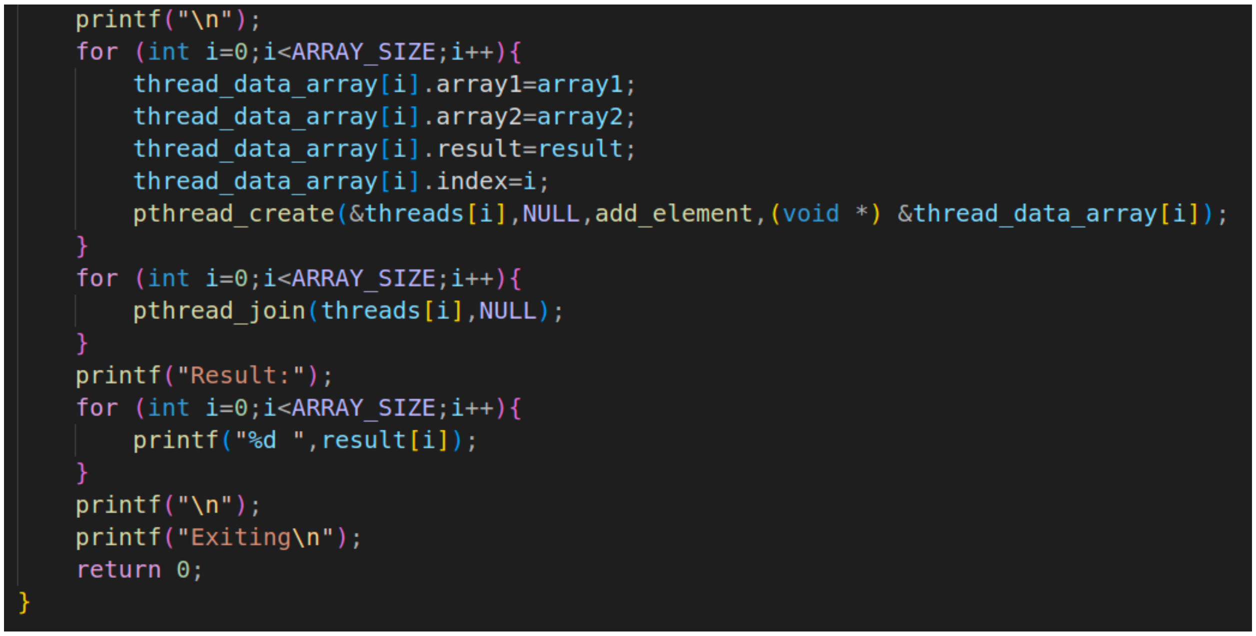
Visually:


task_1.c > main()

```c
#include <pthread.h>
#include <stdio.h>
#define ARRAY_SIZE 5
struct thread_data{
    int *array1;
    int *array2;
    int *result;
    int index;
};
void *add_element(void *threadArg){
    struct thread_data *my_data;
    my_data=(struct thread_data *) threadArg;
    int index=my_data->index;
    my_data->result[index]=my_data->array1[index]+my_data->array2[index];
    pthread_exit(NULL);
}
```

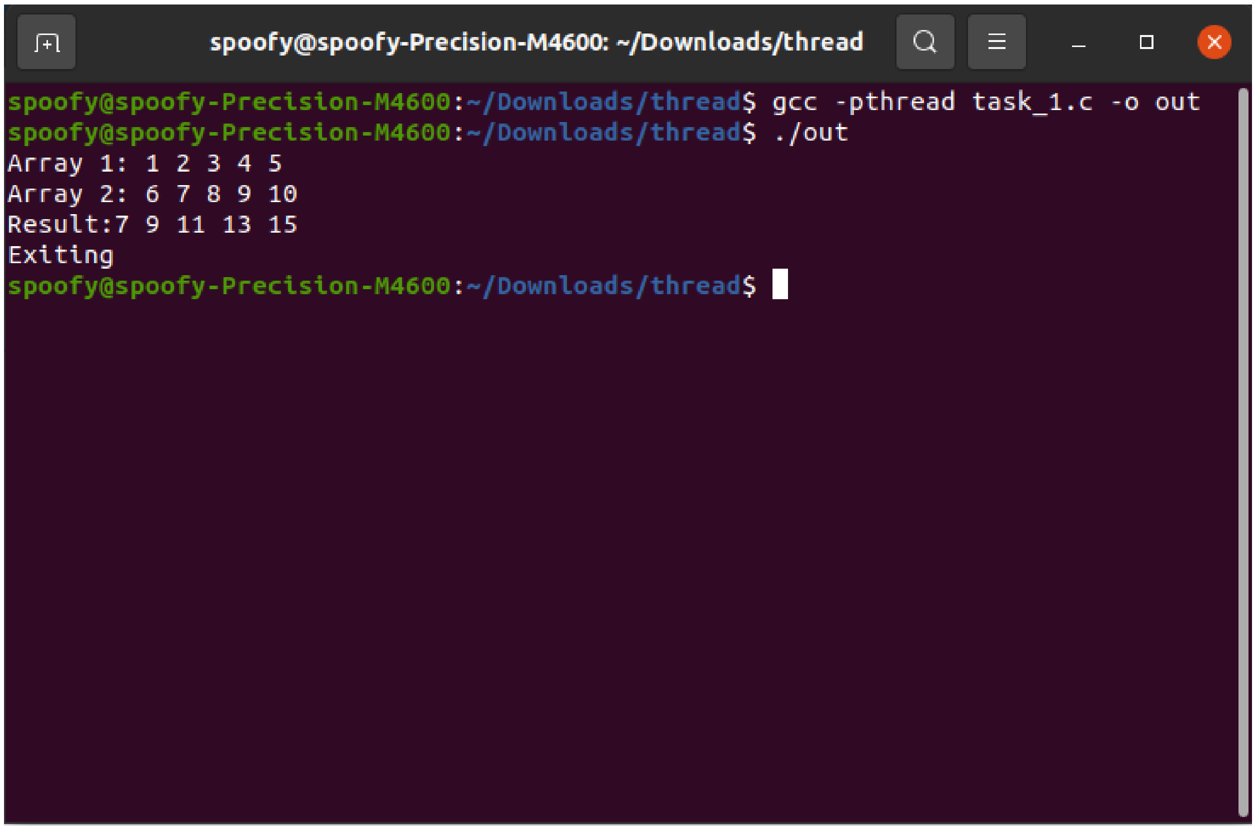


```c
int main(){
    pthread_t threads[ARRAY_SIZE];
    int array1[ARRAY_SIZE]={1,2,3,4,5};
    int array2[ARRAY_SIZE]={6,7,8,9,10};
    int result[ARRAY_SIZE];
    struct thread_data thread_data_array[ARRAY_SIZE];
    printf("Array 1: ");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",array1[i]);
    }
    printf("\n");
    printf("Array 2: ");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",array2[i]);
    }
    printf("\n");
    for (int i=0;i<ARRAY_SIZE;i++){
        thread_data_array[i].array1=array1;
        thread_data_array[i].array2=array2;
        thread_data_array[i].result=result;
        thread_data_array[i].index=i;
        pthread_create(&threads[i],NULL,add_element,(void *) &thread_data_array[i]);
    }
```



```c
    printf("\n");
    for (int i=0;i<ARRAY_SIZE;i++){
        thread_data_array[i].array1=array1;
        thread_data_array[i].array2=array2;
        thread_data_array[i].result=result;
        thread_data_array[i].index=i;
        pthread_create(&threads[i],NULL,add_element,(void *) &thread_data_array[i]);
    }
    for (int i=0;i<ARRAY_SIZE;i++){
        pthread_join(threads[i],NULL);
    }
    printf("Result:");
    for (int i=0;i<ARRAY_SIZE;i++){
        printf("%d ",result[i]);
    }
    printf("\n");
    printf("Exiting\n");
    return 0;
}
```

Running:

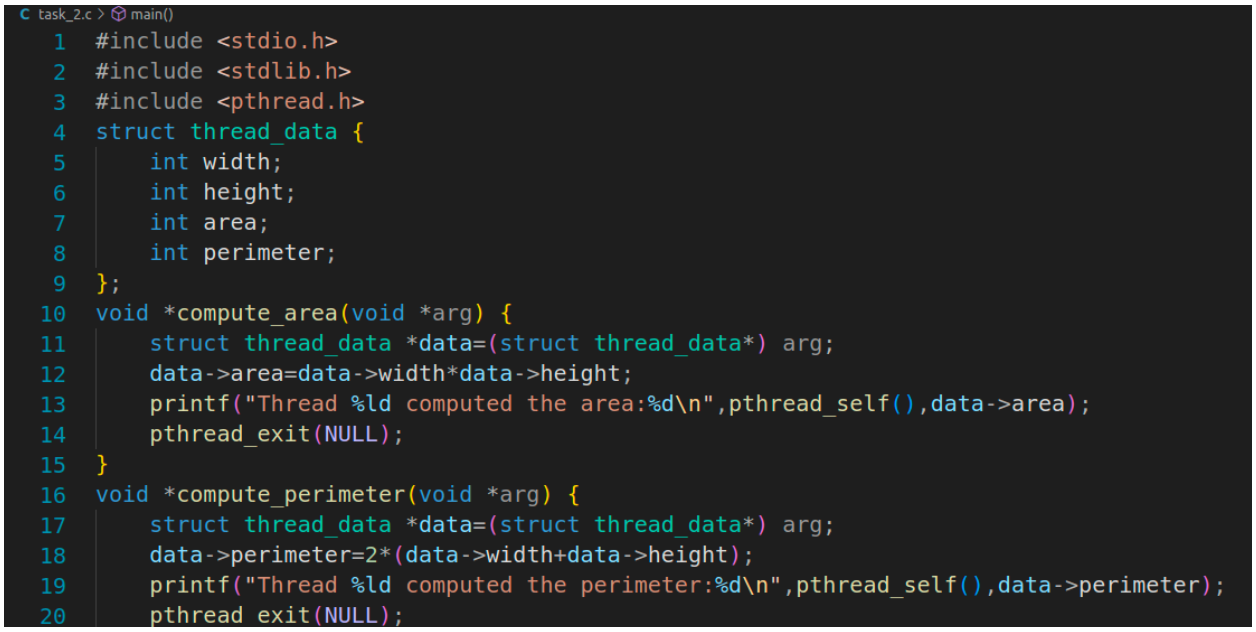


```
spoofy@spoofy-Precision-M4600:~/Downloads/thread$ gcc -pthread task_1.c -o out
spoofy@spoofy-Precision-M4600:~/Downloads/thread$ ./out
Array 1: 1 2 3 4 5
Array 2: 6 7 8 9 10
Result:7 9 11 13 15
Exiting
spoofy@spoofy-Precision-M4600:~/Downloads/thread$ 
```

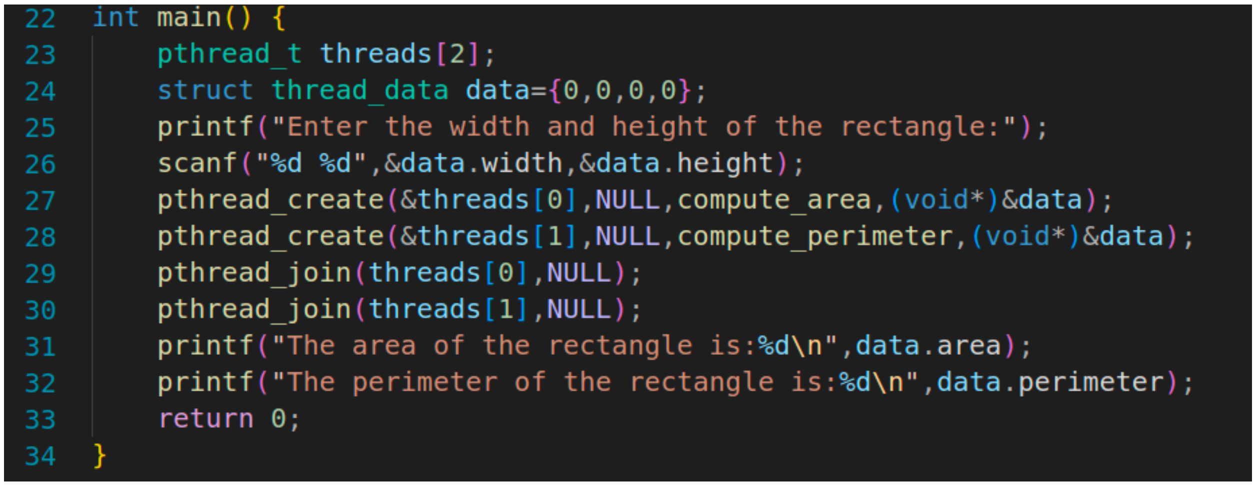
Task 2:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
struct thread_data {
    int width;
    int height;
    int area;
    int perimeter;
};
void *compute_area(void *arg) {
    struct thread_data *data=(struct thread_data*) arg;
    data->area=data->width*data->height;
    printf("Thread %ld computed the area:%d\n",pthread_self(),data->area);
    pthread_exit(NULL);
}
void *compute_perimeter(void *arg) {
    struct thread_data *data=(struct thread_data*) arg;
    data->perimeter=2*(data->width+data->height);
    printf("Thread %ld computed the perimeter:%d\n",pthread_self(),data->perimeter);
    pthread_exit(NULL);
}
int main() {
    pthread_t threads[2];
    struct thread_data data={0,0,0,0};
    printf("Enter the width and height of the rectangle:");
    scanf("%d %d",&data.width,&data.height);
    pthread_create(&threads[0],NULL,compute_area,(void*)&data);
    pthread_create(&threads[1],NULL,compute_perimeter,(void*)&data);
    pthread_join(threads[0],NULL);
    pthread_join(threads[1],NULL);
    printf("The area of the rectangle is:%d\n",data.area);
    printf("The perimeter of the rectangle is:%d\n",data.perimeter);
    return 0;
}
```

Visually:

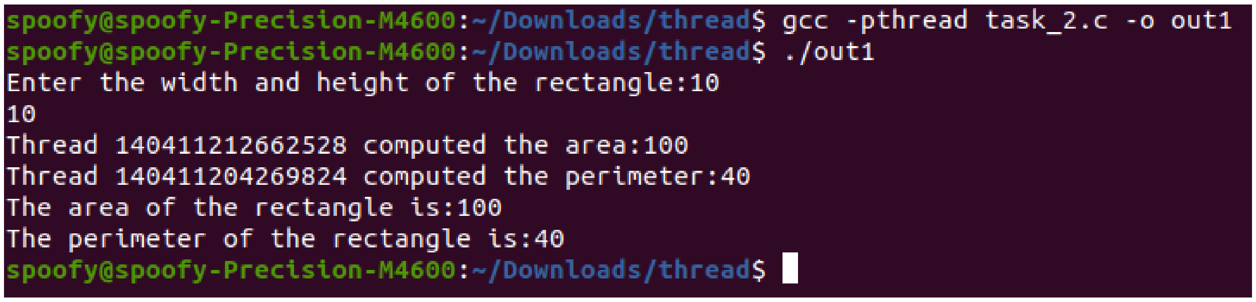


```
task_2.c > main()
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
struct thread_data {
    int width;
    int height;
    int area;
    int perimeter;
};
void *compute_area(void *arg) {
    struct thread_data *data=(struct thread_data*) arg;
    data->area=data->width*data->height;
    printf("Thread %ld computed the area:%d\n",pthread_self(),data->area);
    pthread_exit(NULL);
}
void *compute_perimeter(void *arg) {
    struct thread_data *data=(struct thread_data*) arg;
    data->perimeter=2*(data->width+data->height);
    printf("Thread %ld computed the perimeter:%d\n",pthread_self(),data->perimeter);
    pthread_exit(NULL);
```



```
int main() {
    pthread_t threads[2];
    struct thread_data data={0,0,0,0};
    printf("Enter the width and height of the rectangle:");
    scanf("%d %d",&data.width,&data.height);
    pthread_create(&threads[0],NULL,compute_area,(void*)&data);
    pthread_create(&threads[1],NULL,compute_perimeter,(void*)&data);
    pthread_join(threads[0],NULL);
    pthread_join(threads[1],NULL);
    printf("The area of the rectangle is:%d\n",data.area);
    printf("The perimeter of the rectangle is:%d\n",data.perimeter);
    return 0;
}
```

Running:



```
spoofy@spoofy-Precision-M4600:~/Downloads/thread$ gcc -pthread task_2.c -o out1
spoofy@spoofy-Precision-M4600:~/Downloads/thread$ ./out1
Enter the width and height of the rectangle:10
10
Thread 140411212662528 computed the area:100
Thread 140411204269824 computed the perimeter:40
The area of the rectangle is:100
The perimeter of the rectangle is:40
spoofy@spoofy-Precision-M4600:~/Downloads/thread$
```

FIN.