NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Eman Faisal 23-NTU-CS-1149

SECTION SE: 5th (A)

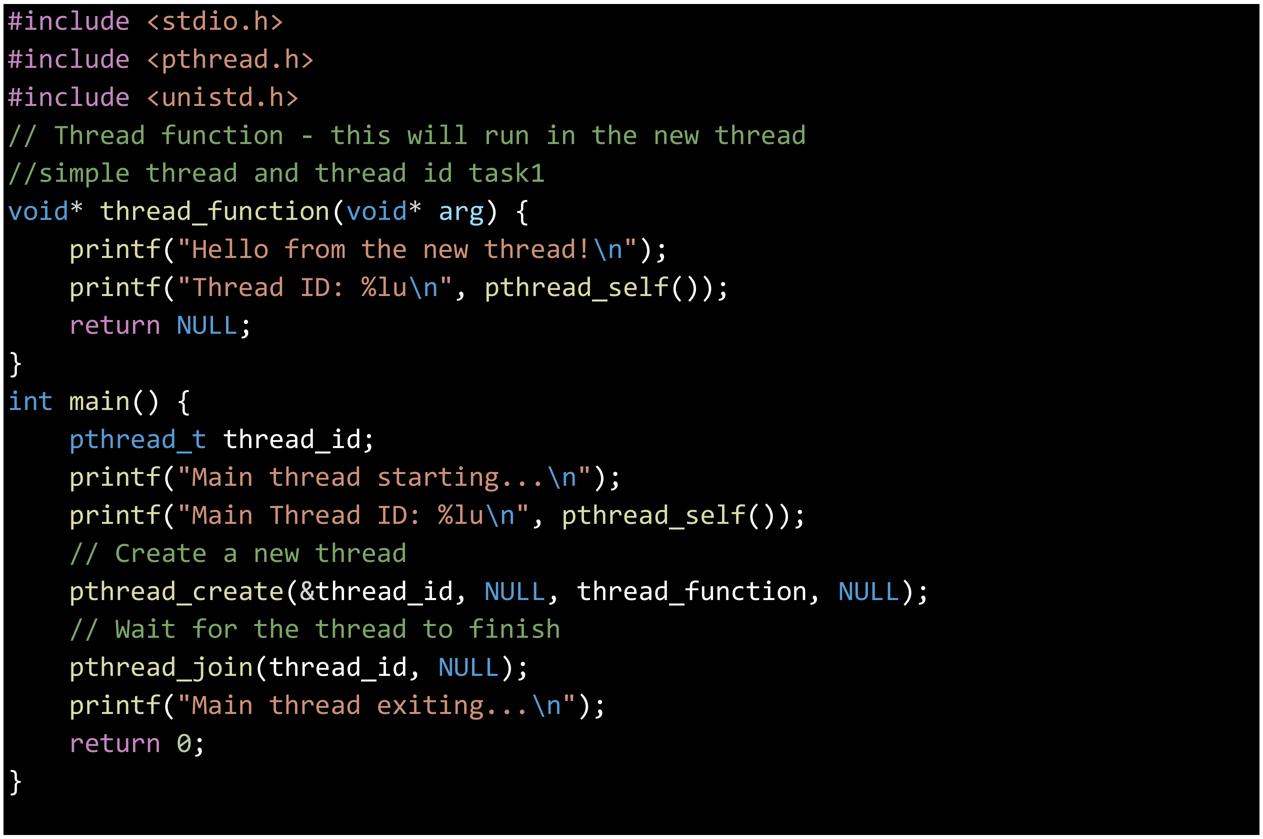
Operating system LAB-5

SUBMITTED TO:

Sir Nasir mahmood

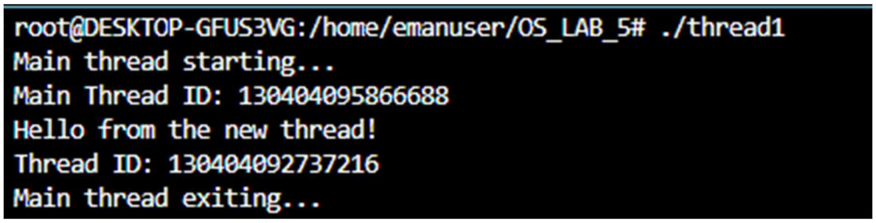
SUBMISSION DATE: 22/9/25

TASK1:



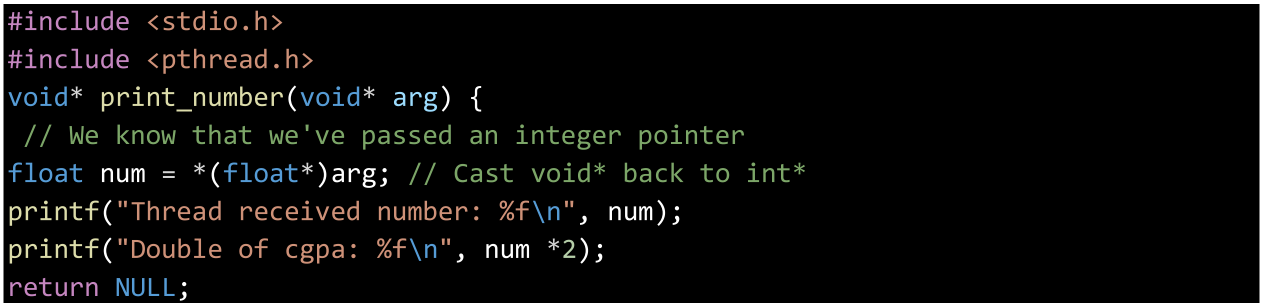
```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
//simple thread and thread id task1
void* thread_function(void* arg) {
    printf("Hello from the new thread!\n");
    printf("Thread ID: %lu\n", pthread_self());
    return NULL;
}
int main() {
    pthread_t thread_id;
    printf("Main thread starting...\n");
    printf("Main Thread ID: %lu\n", pthread_self());
    // Create a new thread
    pthread_create(&thread_id, NULL, thread_function, NULL);
    // Wait for the thread to finish
    pthread_join(thread_id, NULL);
    printf("Main thread exiting...\n");
    return 0;
}
```

OUTPUT:

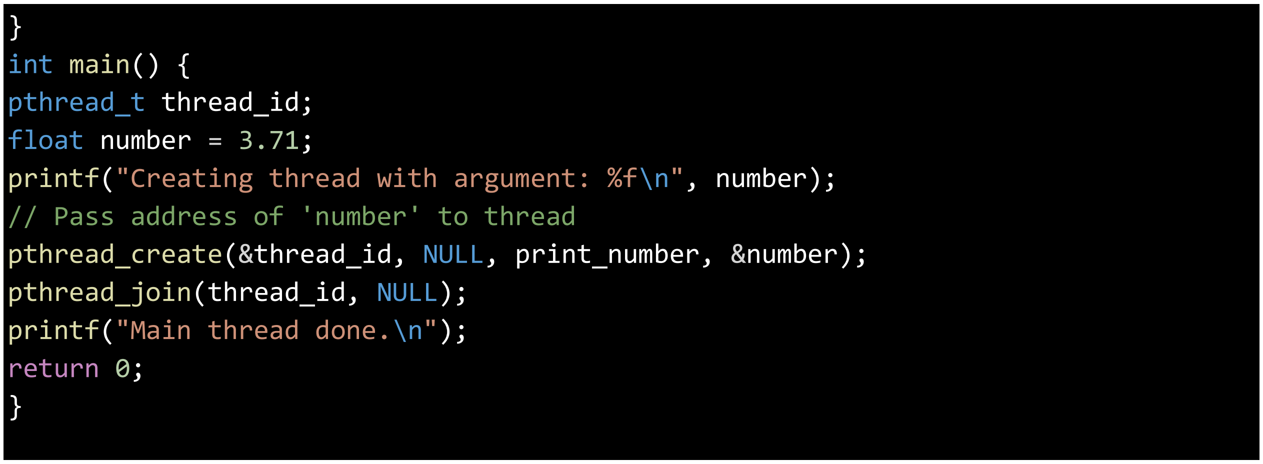


```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread1
Main thread starting...
Main Thread ID: 130404095866688
Hello from the new thread!
Thread ID: 130404092737216
Main thread exiting...
```

TASK2:

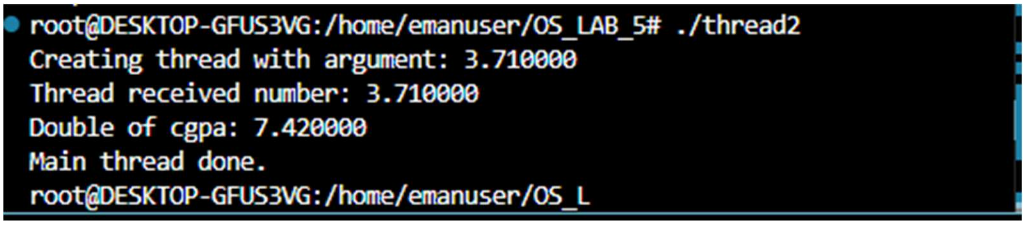


```
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
 // We know that we've passed an integer pointer
float num = *(float*)arg; // Cast void* back to int*
printf("Thread received number: %f\n", num);
printf("Double of cgpa: %f\n", num *2);
return NULL;
```



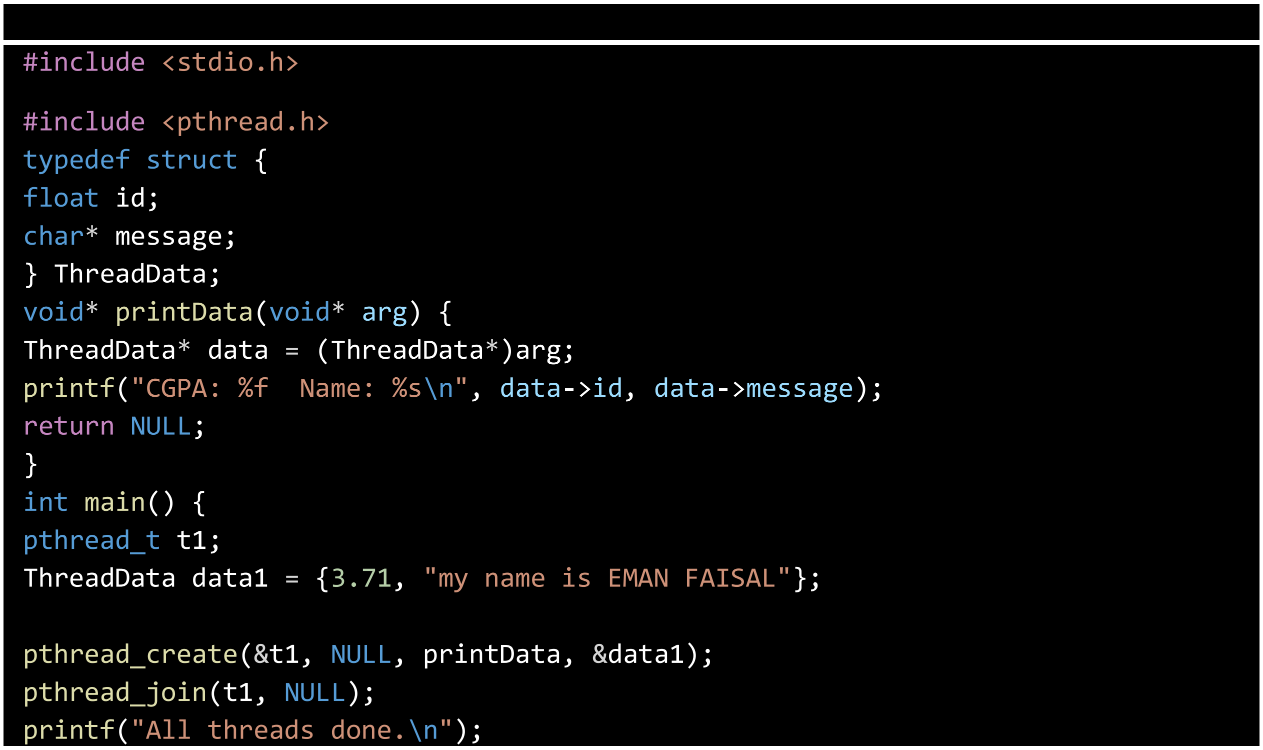
```c
}
int main() {
pthread_t thread_id;
float number = 3.71;
printf("Creating thread with argument: %f\n", number);
// Pass address of 'number' to thread
pthread_create(&thread_id, NULL, print_number, &number);
pthread_join(thread_id, NULL);
printf("Main thread done.\n");
return 0;
}
```

OUTPUT:



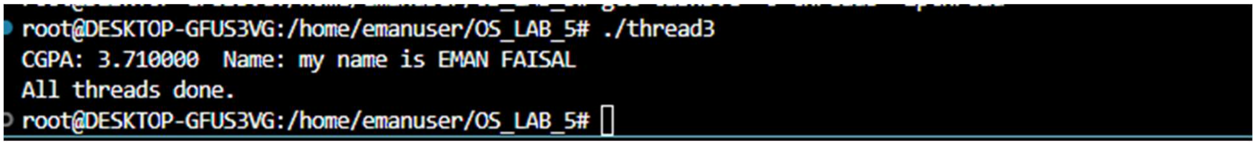
```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread2
Creating thread with argument: 3.710000
Thread received number: 3.710000
Double of cgpa: 7.420000
Main thread done.
root@DESKTOP-GFUS3VG:/home/emanuser/OS_L
```

TASK3:



```c
#include <stdio.h>

#include <pthread.h>
typedef struct {
float id;
char* message;
} ThreadData;
void* printData(void* arg) {
ThreadData* data = (ThreadData*)arg;
printf("CGPA: %f  Name: %s\n", data->id, data->message);
return NULL;
}
int main() {
pthread_t t1;
ThreadData data1 = {3.71, "my name is EMAN FAISAL"};

pthread_create(&t1, NULL, printData, &data1);
pthread_join(t1, NULL);
printf("All threads done.\n");
return 0;
}
```

OUTPUT:

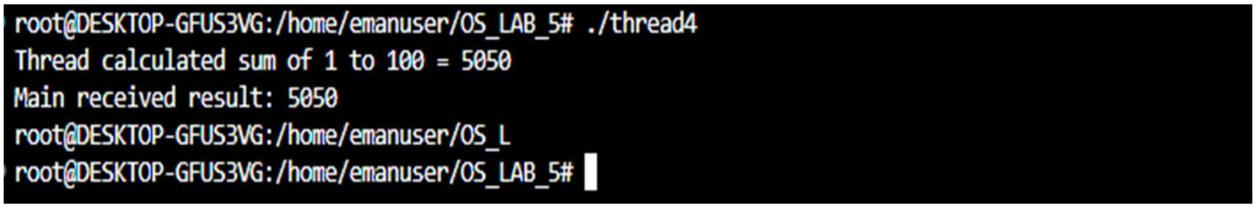


```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread3
CGPA: 3.710000  Name: my name is EMAN FAISAL
All threads done.
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# 
```

TASK_4:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
int n = *(int*)arg;
int* result = malloc(sizeof(int)); // Allocate memory for result
*result = 0;
for (int i = 1; i <= n; i++) {
*result += i;
}
printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
return (void*)result; // Return the result
}
int main() {
pthread_t thread_id;
int n = 100;
void* sum;
pthread_create(&thread_id, NULL, calculate_sum, &n);
// Get the return value from thread
pthread_join(thread_id, &sum);
printf("Main received result: %d\n", *(int*)sum);
free(sum); // Don't forget to free allocated memory
return 0;
}
```

OUTPUT:

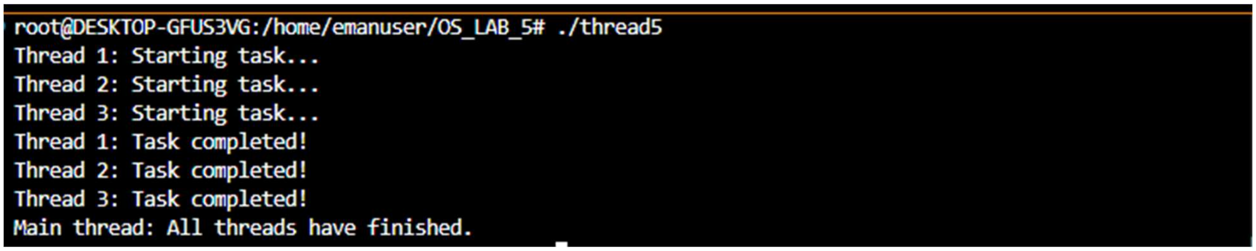


```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread4
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
root@DESKTOP-GFUS3VG:/home/emanuser/OS_L
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5#
```

TASK5:

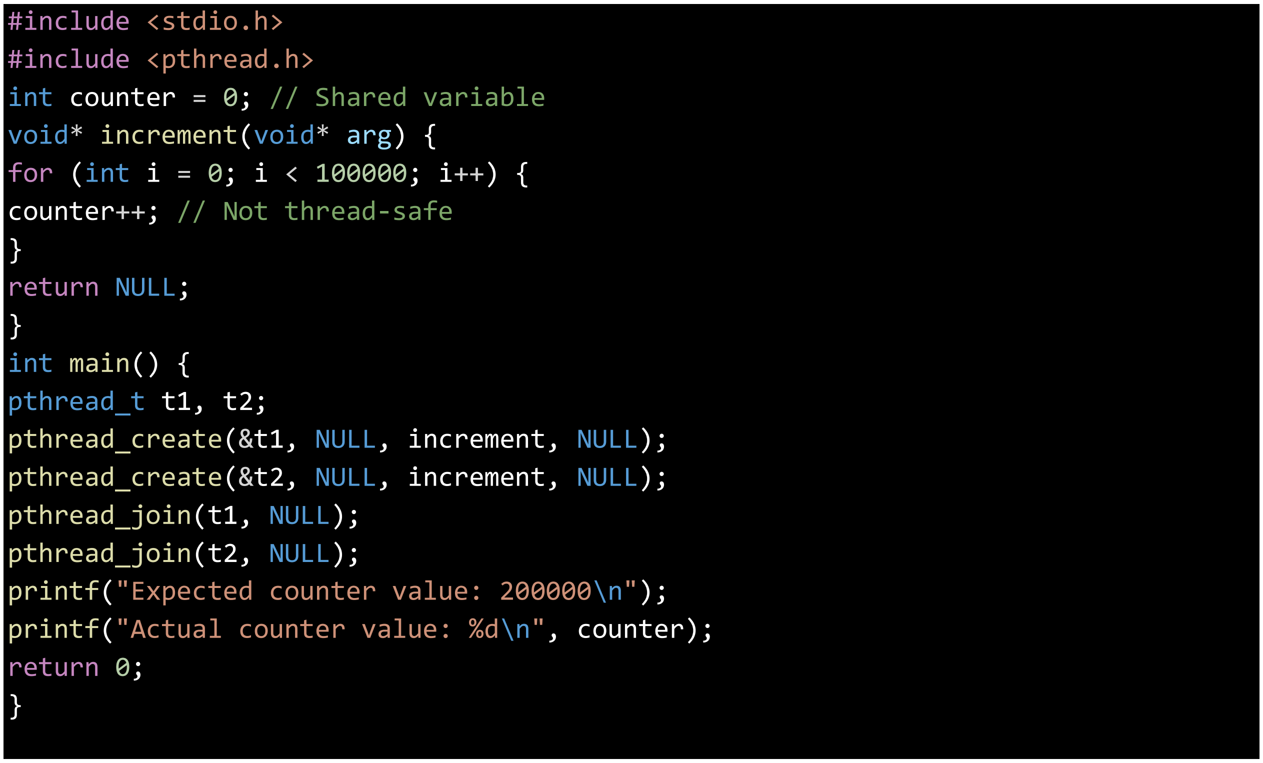
```
#include <stdio.h>

#include <pthread.h>
#include <unistd.h>
void* worker(void* arg) {
int thread_num = *(int*)arg;
printf("Thread %d: Starting task...\n", thread_num);
sleep(1); // Simulate some work
printf("Thread %d: Task completed!\n", thread_num);
return NULL;
}
int main() {
pthread_t threads[3];
int thread_ids[3];
for (int i = 0; i < 3; i++) {
thread_ids[i] = i + 1;
pthread_create(&threads[i], NULL, worker, &thread_ids[i]);
}
for (int i = 0; i < 3; i++) {
    pthread_join(threads[i], NULL);
}
printf("Main thread: All threads have finished.\n");
return 0;
}
```

OUTPUT:



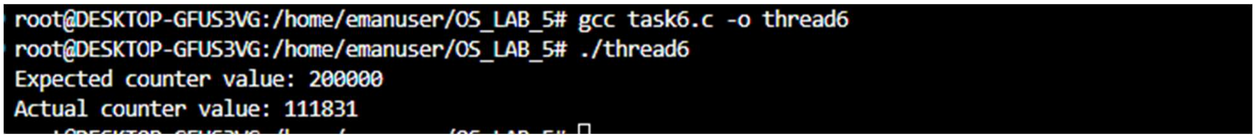
```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread5
Thread 1: Starting task...
Thread 2: Starting task...
Thread 3: Starting task...
Thread 1: Task completed!
Thread 2: Task completed!
Thread 3: Task completed!
Main thread: All threads have finished.
```

TASK6:



```c
#include <stdio.h>
#include <pthread.h>
int counter = 0; // Shared variable
void* increment(void* arg) {
for (int i = 0; i < 100000; i++) {
counter++; // Not thread-safe
}
return NULL;
}
int main() {
pthread_t t1, t2;
pthread_create(&t1, NULL, increment, NULL);
pthread_create(&t2, NULL, increment, NULL);
pthread_join(t1, NULL);
pthread_join(t2, NULL);
printf("Expected counter value: 200000\n");
printf("Actual counter value: %d\n", counter);
return 0;
}
```

OUTPUT:



```
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# gcc task6.c -o thread6
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# ./thread6
Expected counter value: 200000
Actual counter value: 111831
root@DESKTOP-GFUS3VG:/home/emanuser/OS_LAB_5# 
```