National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir

Submitted by:

Esha Mubashir Khan

Reg number:

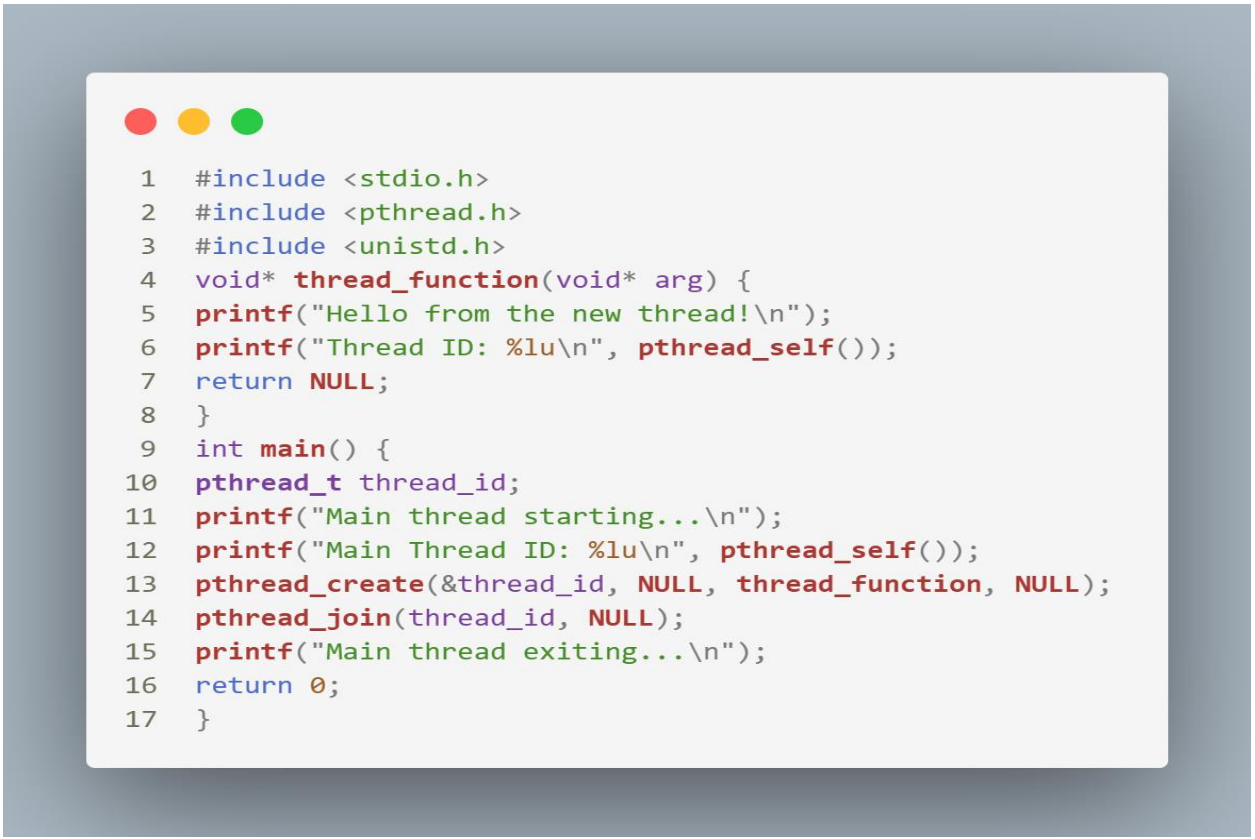
23-NTU-CS-1151

Lab no: **5**

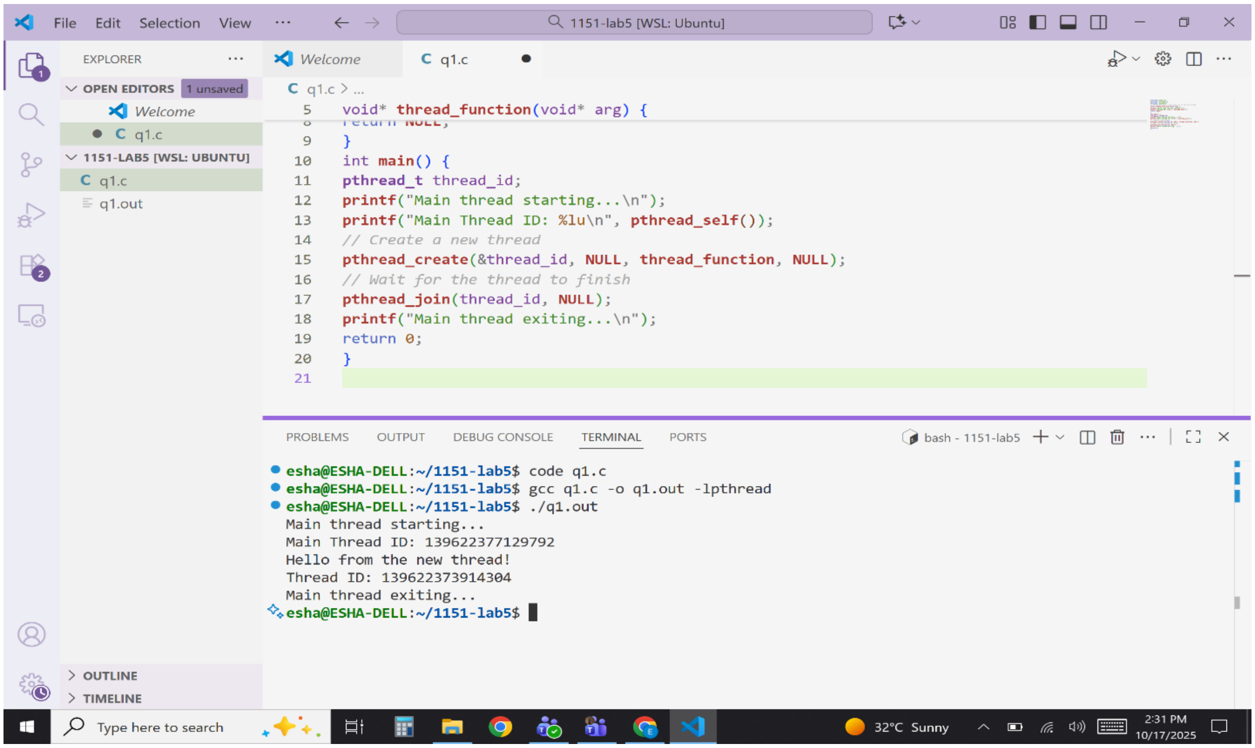
Semester: **5th**

3. C Programs with Threads

Program 1: Creating a Simple Thread



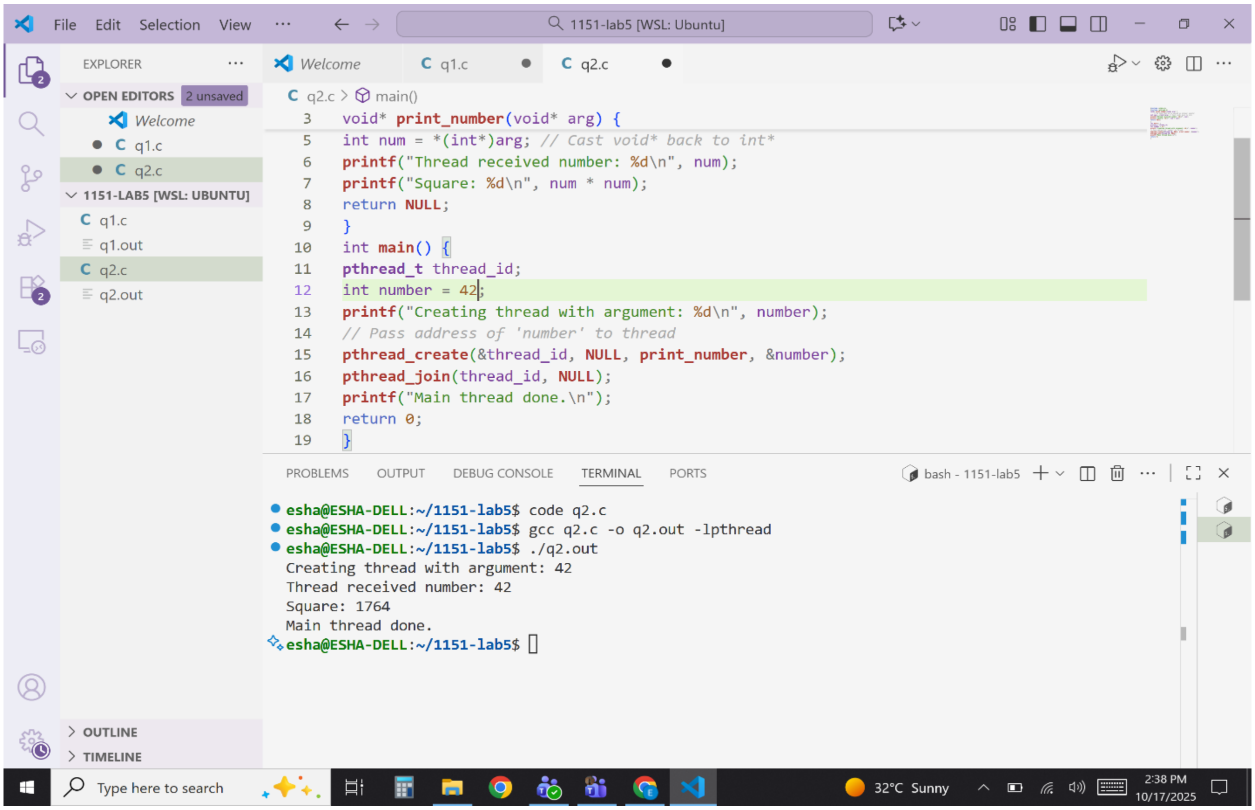
```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* thread_function(void* arg) {
printf("Hello from the new thread!\n");
printf("Thread ID: %lu\n", pthread_self());
return NULL;
}
int main() {
pthread_t thread_id;
printf("Main thread starting...\n");
printf("Main Thread ID: %lu\n", pthread_self());
pthread_create(&thread_id, NULL, thread_function, NULL);
pthread_join(thread_id, NULL);
printf("Main thread exiting...\n");
return 0;
}
```



```c
void* thread_function(void* arg) {
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;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ code q1.c
esha@ESHA-DELL:~/1151-lab5$ gcc q1.c -o q1.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q1.out
Main thread starting...
Main Thread ID: 139622377129792
Hello from the new thread!
Thread ID: 139622373914304
Main thread exiting...
esha@ESHA-DELL:~/1151-lab5$
```

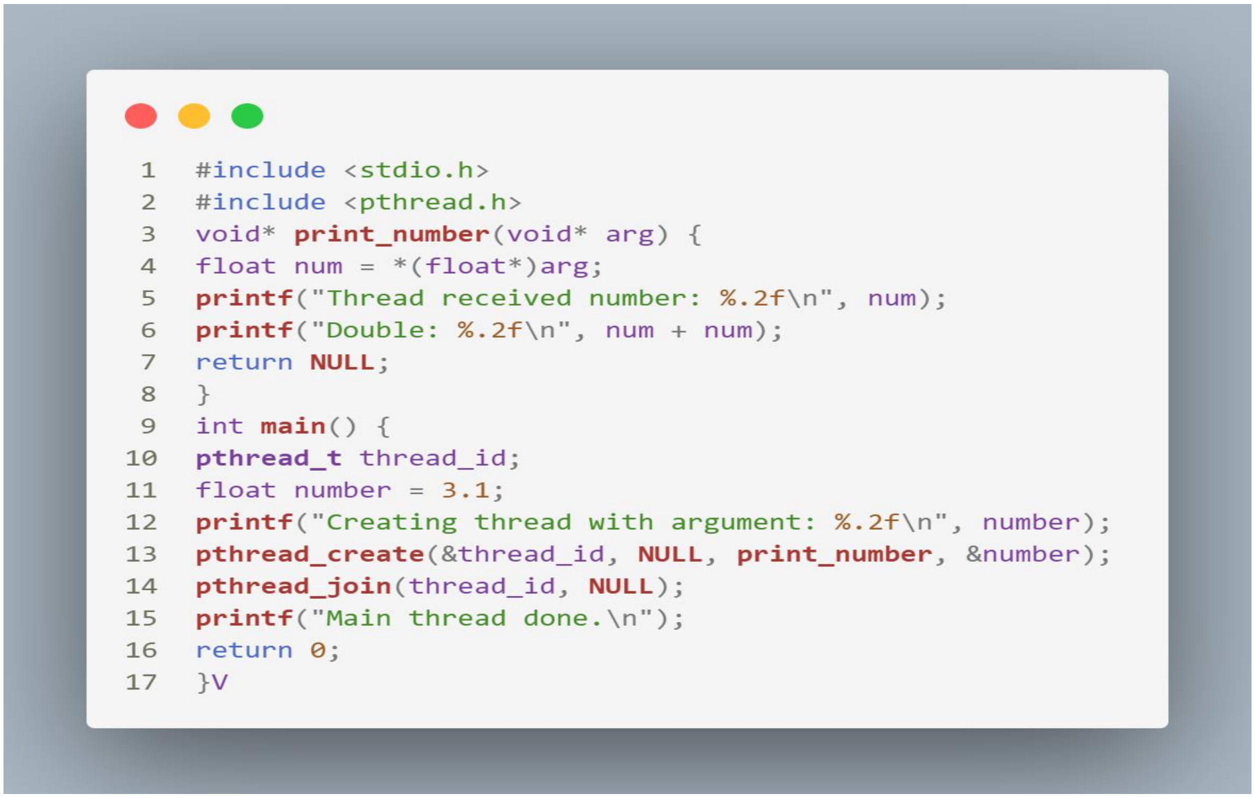
Program 2: Passing Arguments to Threads



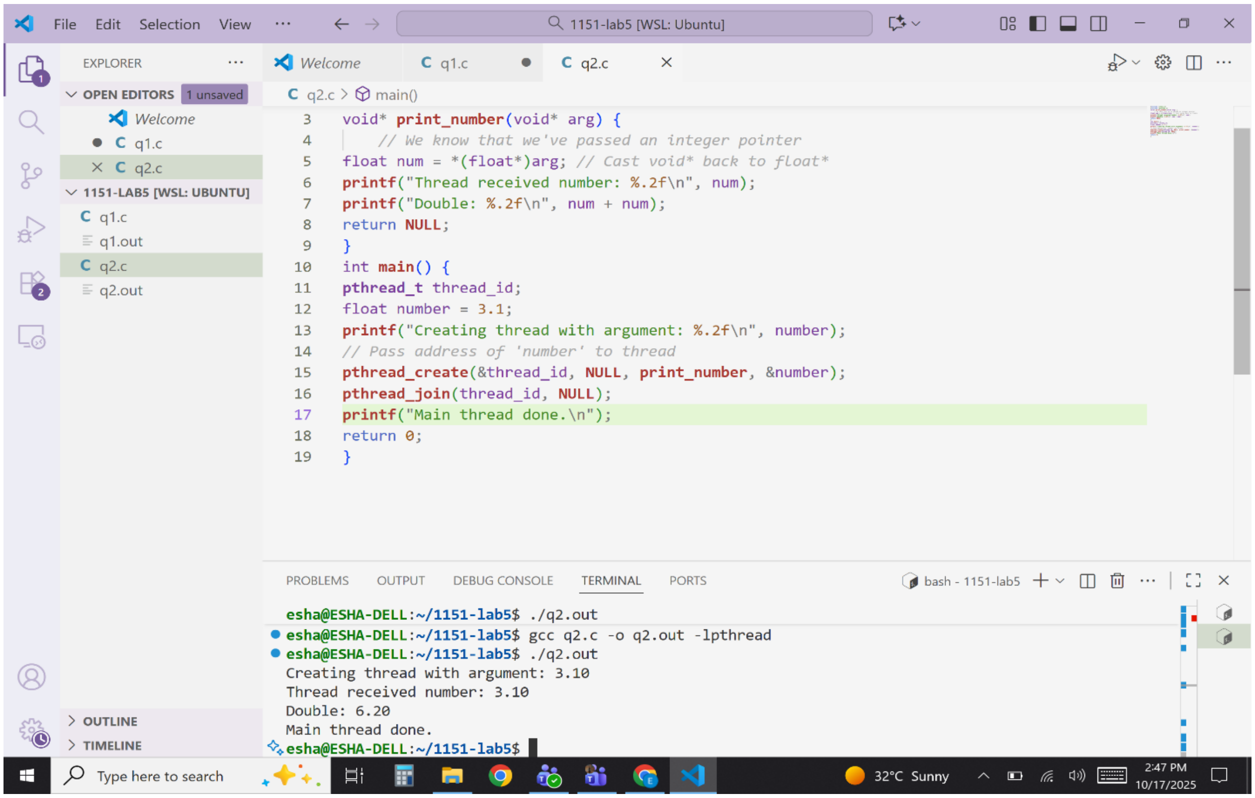
```c
void* print_number(void* arg) {
int num = *(int*)arg; // Cast void* back to int*
printf("Thread received number: %d\n", num);
printf("Square: %d\n", num * num);
return NULL;
}
int main() {
pthread_t thread_id;
int number = 42;
printf("Creating thread with argument: %d\n", number);
// Pass address of 'number' to thread
pthread_create(&thread_id, NULL, print_number, &number);
pthread_join(thread_id, NULL);
printf("Main thread done.\n");
return 0;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ code q2.c
esha@ESHA-DELL:~/1151-lab5$ gcc q2.c -o q2.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
esha@ESHA-DELL:~/1151-lab5$
```

PRINT CGPA:



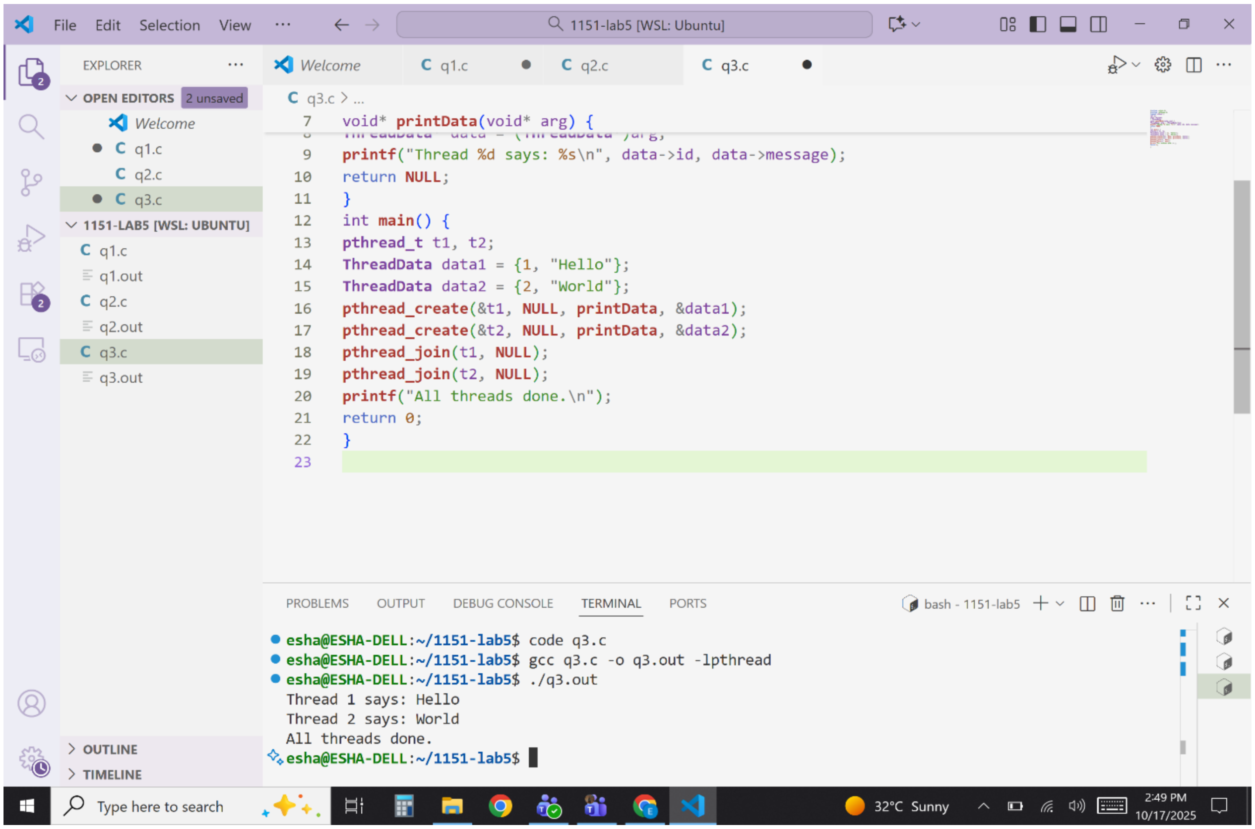
```c
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
float num = *(float*)arg;
printf("Thread received number: %.2f\n", num);
printf("Double: %.2f\n", num + num);
return NULL;
}
int main() {
pthread_t thread_id;
float number = 3.1;
printf("Creating thread with argument: %.2f\n", number);
pthread_create(&thread_id, NULL, print_number, &number);
pthread_join(thread_id, NULL);
printf("Main thread done.\n");
return 0;
}V
```



```c
void* print_number(void* arg) {
    // We know that we've passed an integer pointer
float num = *(float*)arg; // Cast void* back to float*
printf("Thread received number: %.2f\n", num);
printf("Double: %.2f\n", num + num);
return NULL;
}
int main() {
pthread_t thread_id;
float number = 3.1;
printf("Creating thread with argument: %.2f\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;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
esha@ESHA-DELL:~/1151-lab5$ gcc q2.c -o q2.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
Creating thread with argument: 3.10
Thread received number: 3.10
Double: 6.20
Main thread done.
esha@ESHA-DELL:~/1151-lab5$
```

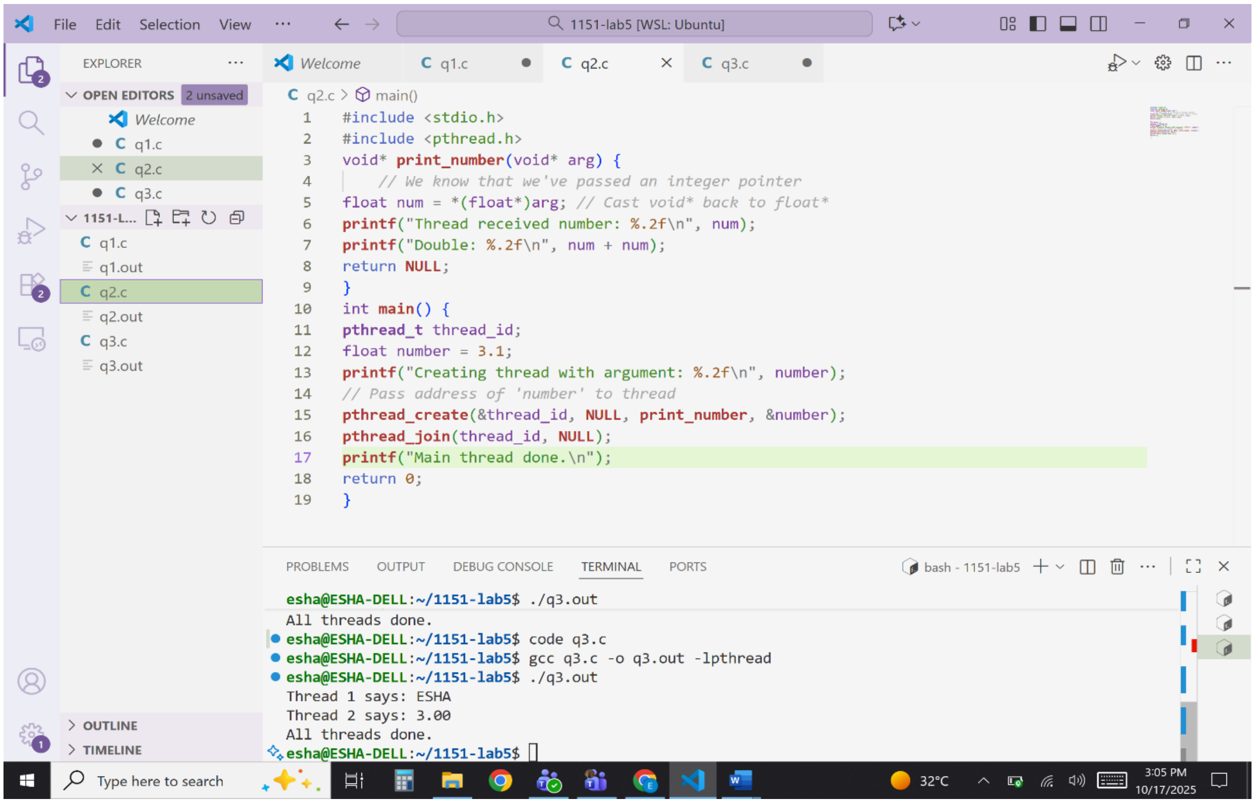
Program 3: Passing Multiple Data



```c
void* printData(void* arg) {
ThreadData* data = (ThreadData*)arg;
printf("Thread %d says: %s\n", data->id, data->message);
return NULL;
}
int main() {
pthread_t t1, t2;
ThreadData data1 = {1, "Hello"};
ThreadData data2 = {2, "World"};
pthread_create(&t1, NULL, printData, &data1);
pthread_create(&t2, NULL, printData, &data2);
pthread_join(t1, NULL);
pthread_join(t2, NULL);
printf("All threads done.\n");
return 0;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ code q3.c
esha@ESHA-DELL:~/1151-lab5$ gcc q3.c -o q3.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
Thread 1 says: Hello
Thread 2 says: World
All threads done.
esha@ESHA-DELL:~/1151-lab5$
```

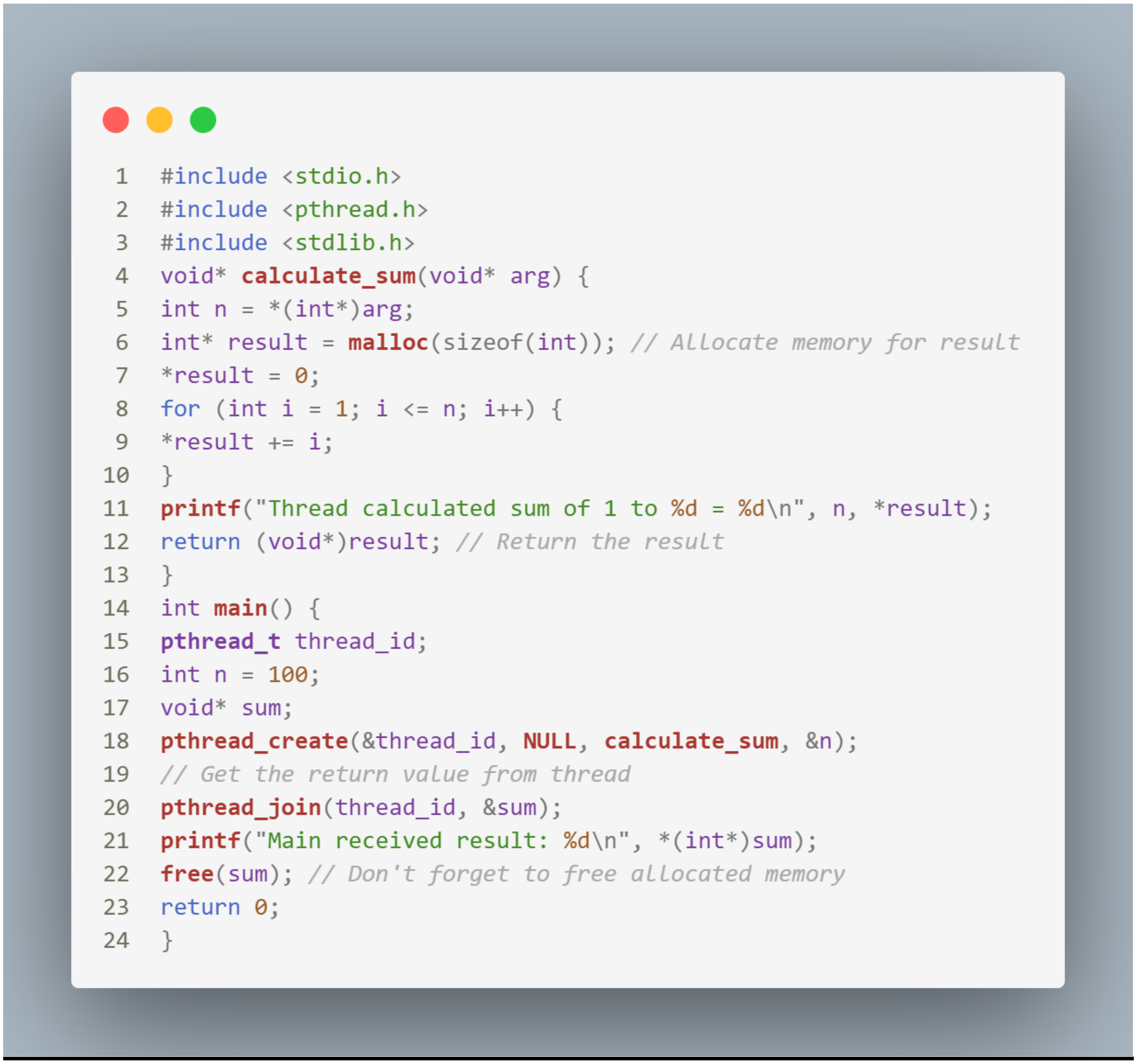
PRINT NAME & CGPA:



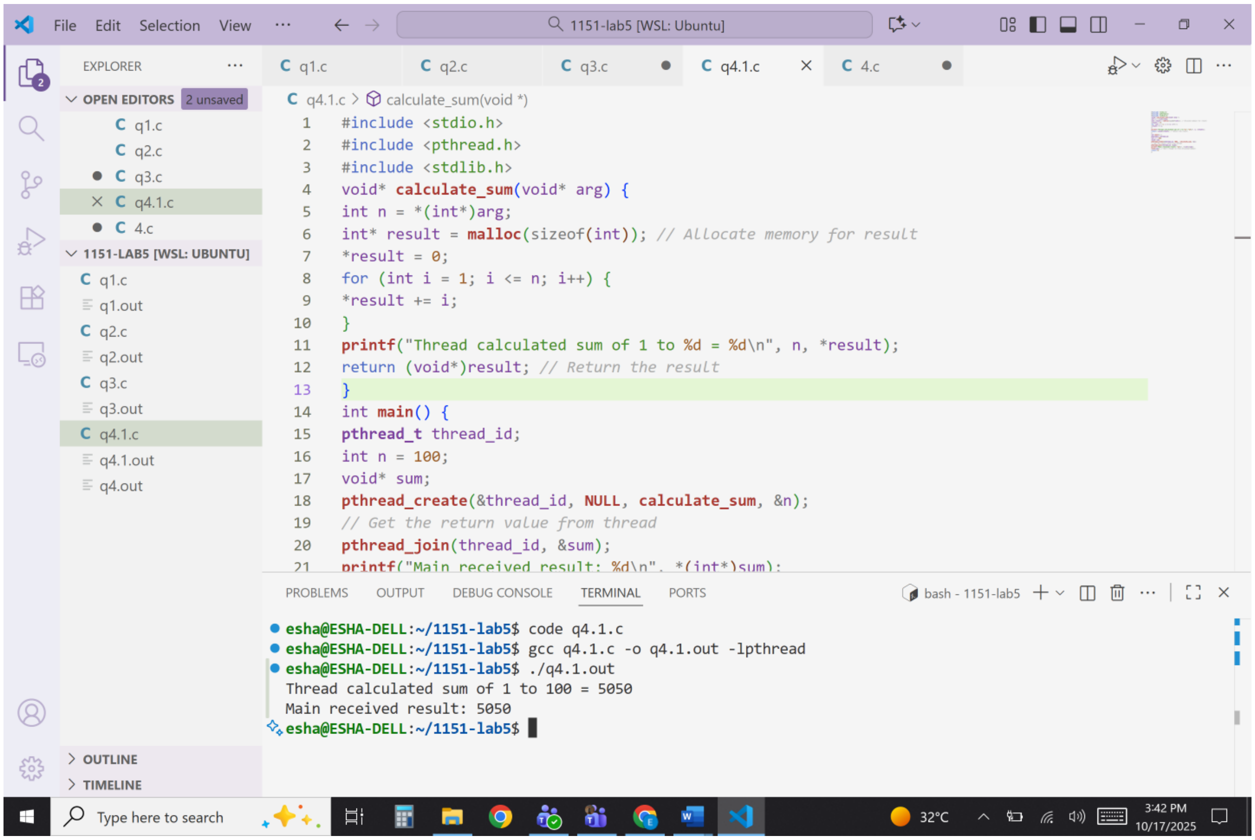
```c
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
    // We know that we've passed an integer pointer
float num = *(float*)arg; // Cast void* back to float*
printf("Thread received number: %.2f\n", num);
printf("Double: %.2f\n", num + num);
return NULL;
}
int main() {
pthread_t thread_id;
float number = 3.1;
printf("Creating thread with argument: %.2f\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;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
All threads done.
esha@ESHA-DELL:~/1151-lab5$ code q3.c
esha@ESHA-DELL:~/1151-lab5$ gcc q3.c -o q3.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
Thread 1 says: ESHA
Thread 2 says: 3.00
All threads done.
esha@ESHA-DELL:~/1151-lab5$
```

Program 4.1: Thread Return Values



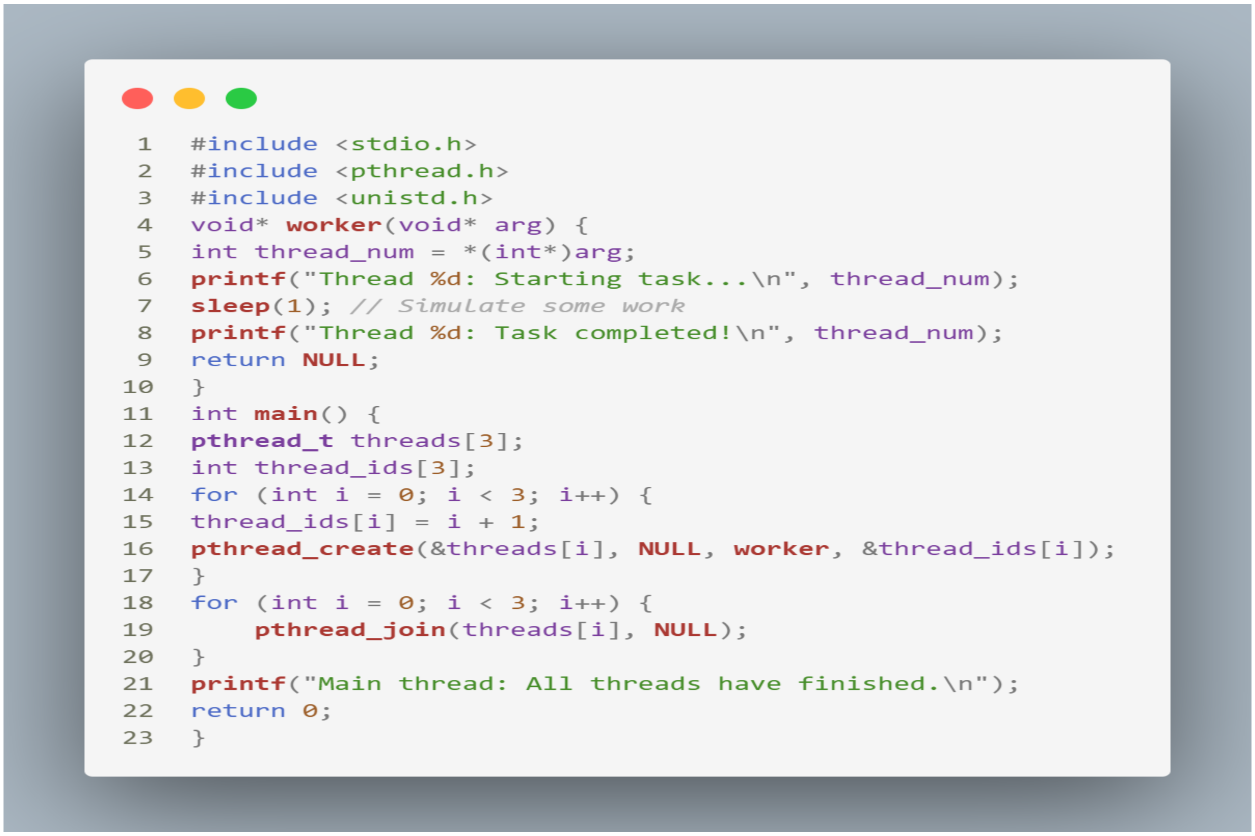
```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
int n = *(int*)arg;
int* result = malloc(sizeof(int)); // Allocate memory for result
*result = 0;
for (int i = 1; i <= n; i++) {
*result += i;
}
printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
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;
}
```



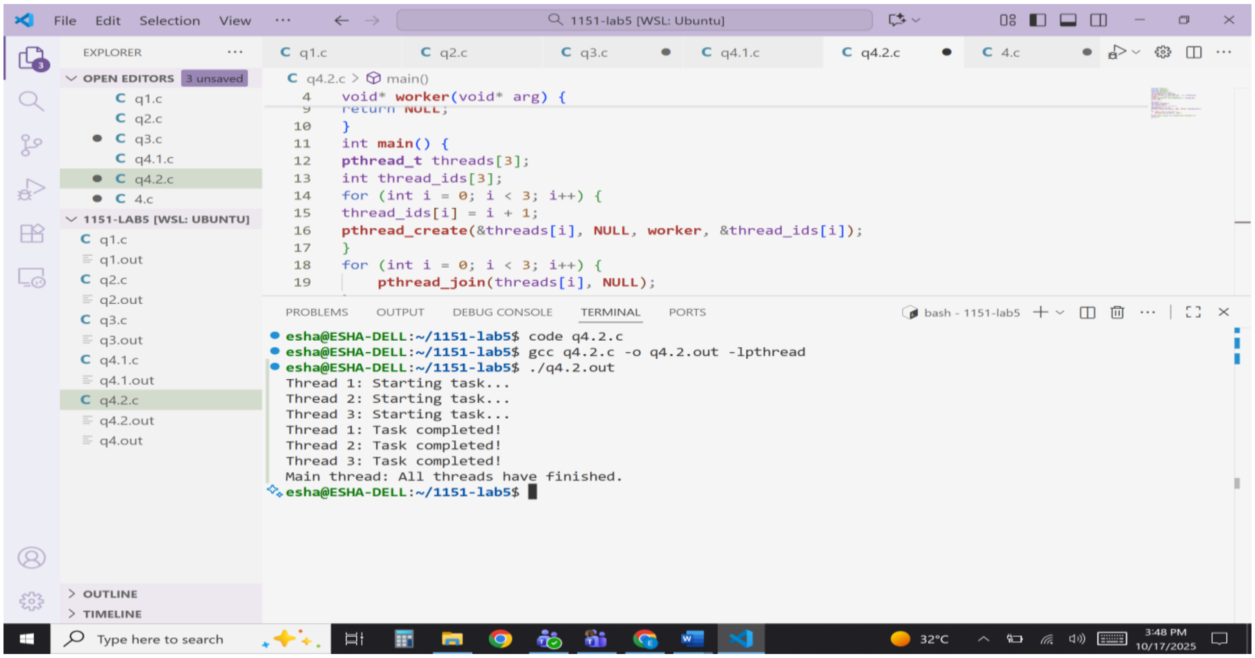
```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
int n = *(int*)arg;
int* result = malloc(sizeof(int)); // Allocate memory for result
*result = 0;
for (int i = 1; i <= n; i++) {
*result += i;
}
printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
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);
```

```
esha@ESHA-DELL:~/1151-lab5$ code q4.1.c
esha@ESHA-DELL:~/1151-lab5$ gcc q4.1.c -o q4.1.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q4.1.out
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
esha@ESHA-DELL:~/1151-lab5$
```

Program 1: Creating and Running Multiple Threads



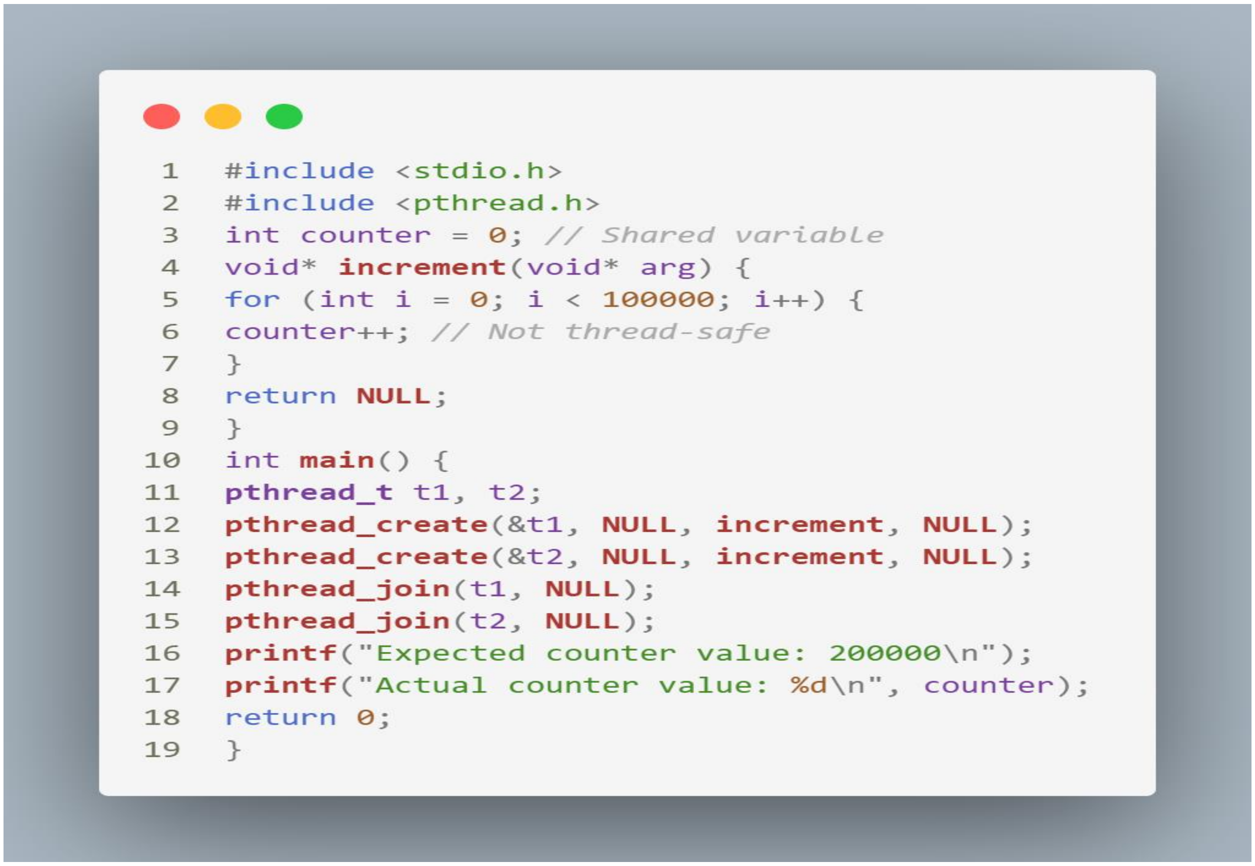
```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* worker(void* arg) {
int thread_num = *(int*)arg;
printf("Thread %d: Starting task...\n", thread_num);
sleep(1); // Simulate some work
printf("Thread %d: Task completed!\n", thread_num);
return NULL;
}
int main() {
pthread_t threads[3];
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;
}
```



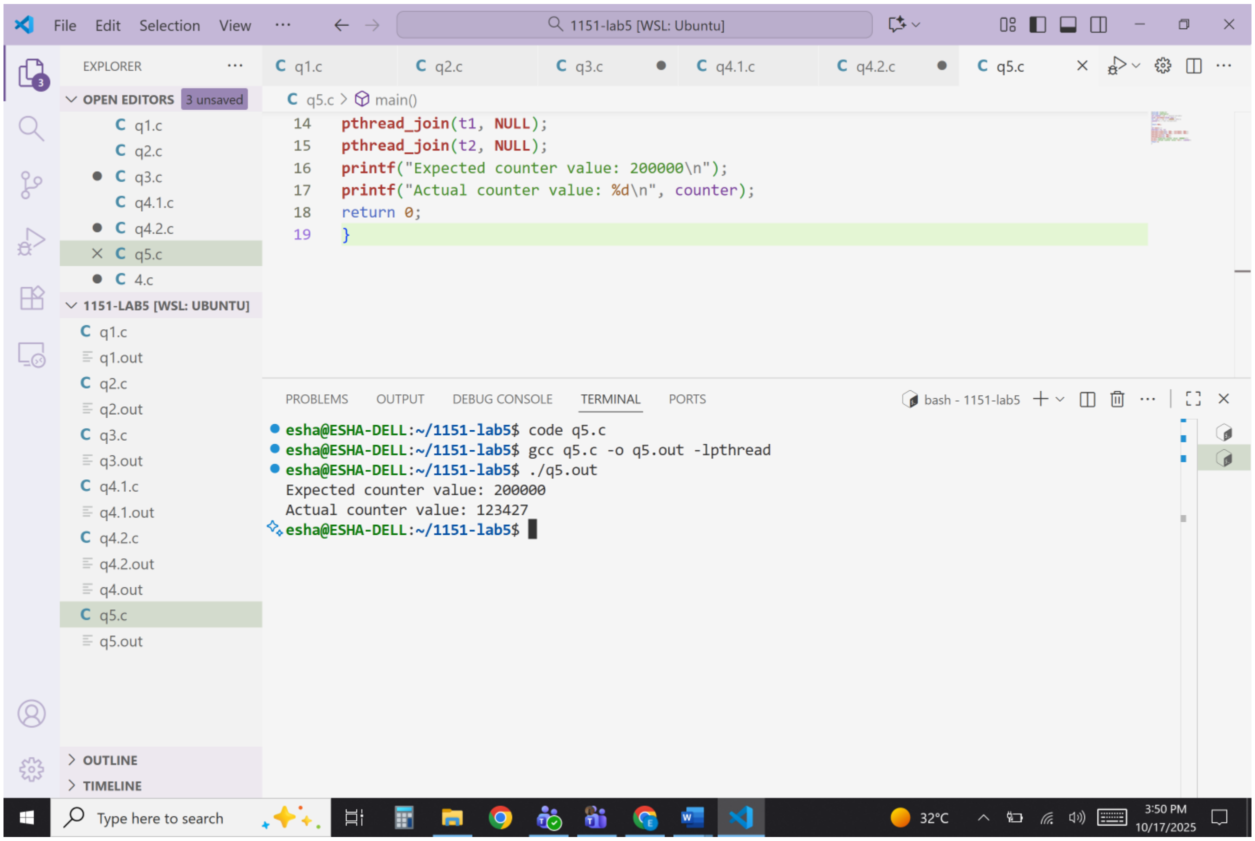
```c
void* worker(void* arg) {
return NULL;
}
int main() {
pthread_t threads[3];
int thread_ids[3];
for (int i = 0; i < 3; i++) {
thread_ids[i] = i + 1;
pthread_create(&threads[i], NULL, worker, &thread_ids[i]);
}
for (int i = 0; i < 3; i++) {
    pthread_join(threads[i], NULL);
```

```
esha@ESHA-DELL:~/1151-lab5$ code q4.2.c
esha@ESHA-DELL:~/1151-lab5$ gcc q4.2.c -o q4.2.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q4.2.out
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.
esha@ESHA-DELL:~/1151-lab5$
```

Program 2: Demonstrating a Race Condition



```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;
}
```



```c
pthread_join(t1, NULL);
pthread_join(t2, NULL);
printf("Expected counter value: 200000\n");
printf("Actual counter value: %d\n", counter);
return 0;
}
```

```
esha@ESHA-DELL:~/1151-lab5$ code q5.c
esha@ESHA-DELL:~/1151-lab5$ gcc q5.c -o q5.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q5.out
Expected counter value: 200000
Actual counter value: 123427
esha@ESHA-DELL:~/1151-lab5$
```