National Textile University

Department of Computer Science

Subject:

Operating Systems

Submitted To:

Sir Nasir Mehmood

Submitted By:

Alishba Riasat

Registration No:

23-NTU-CS-1135

Lab No:

5

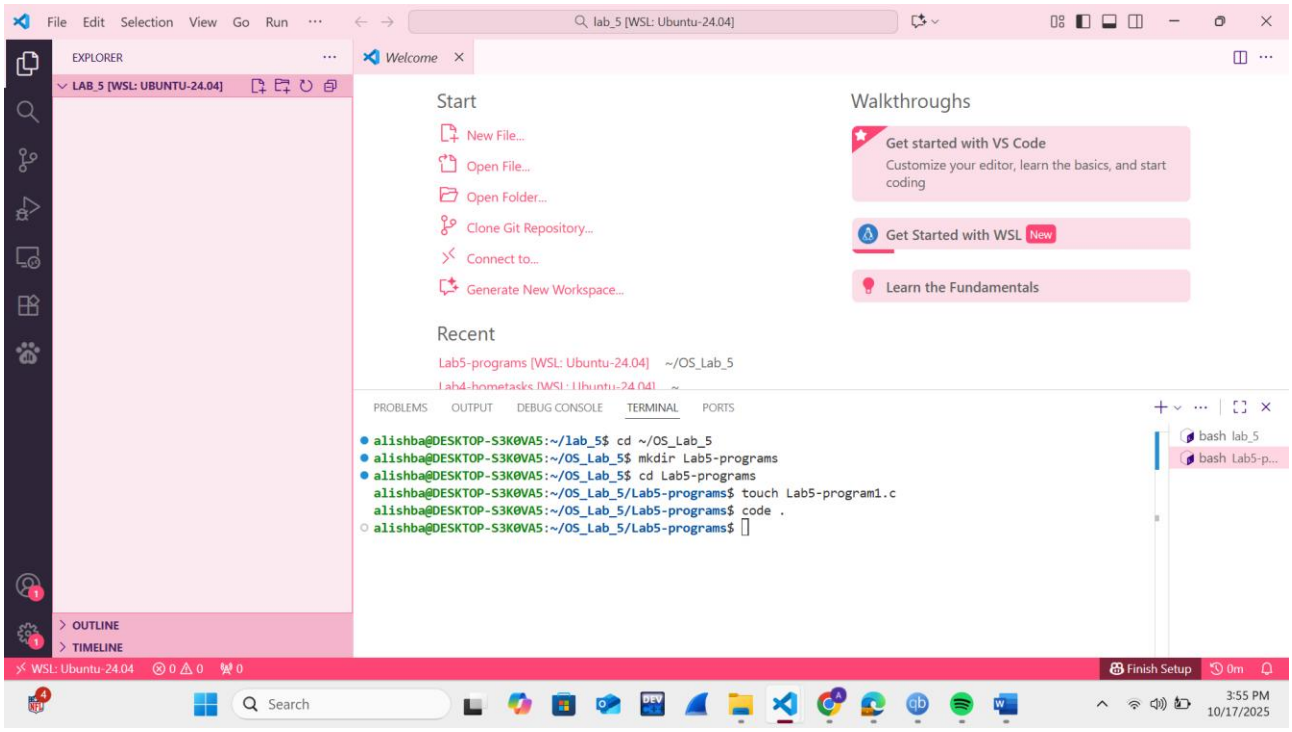
Semester:

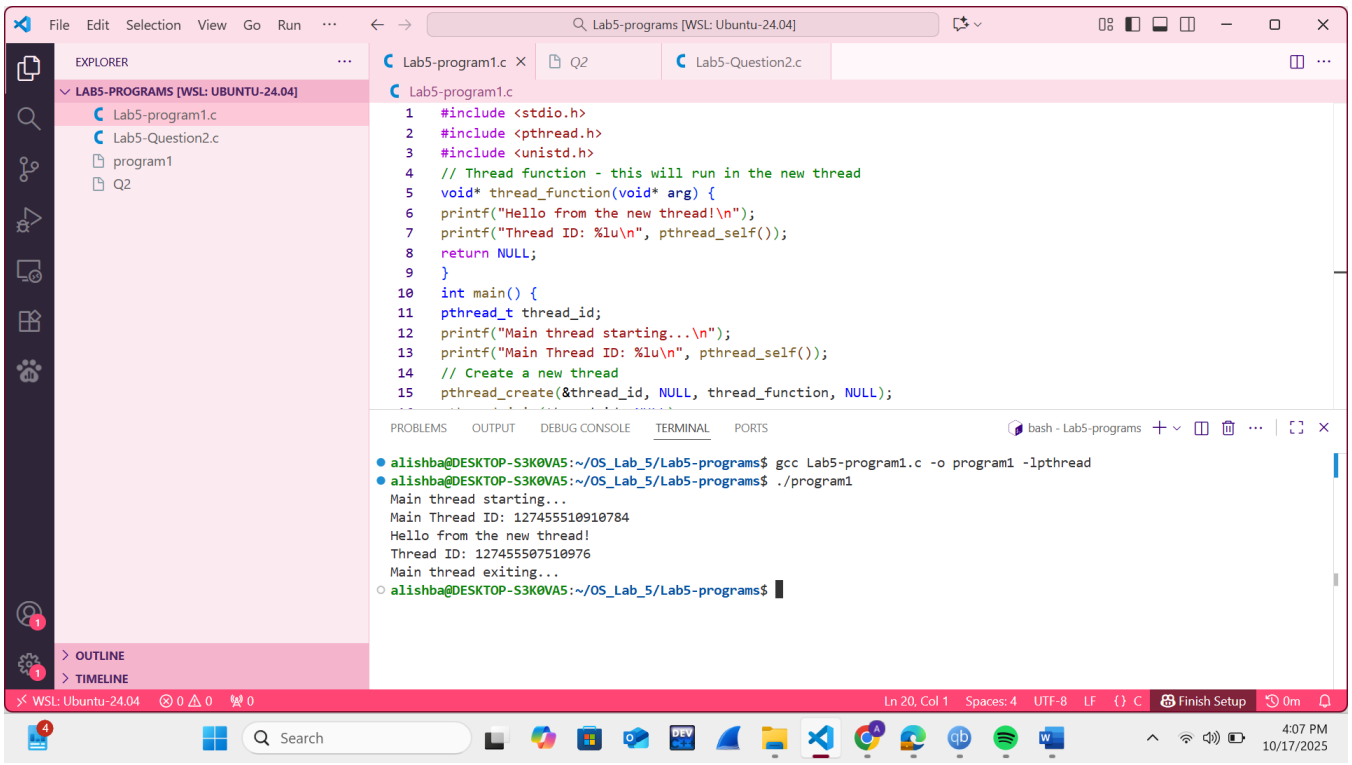
5th

3. C Programs with Threads

Program 1:

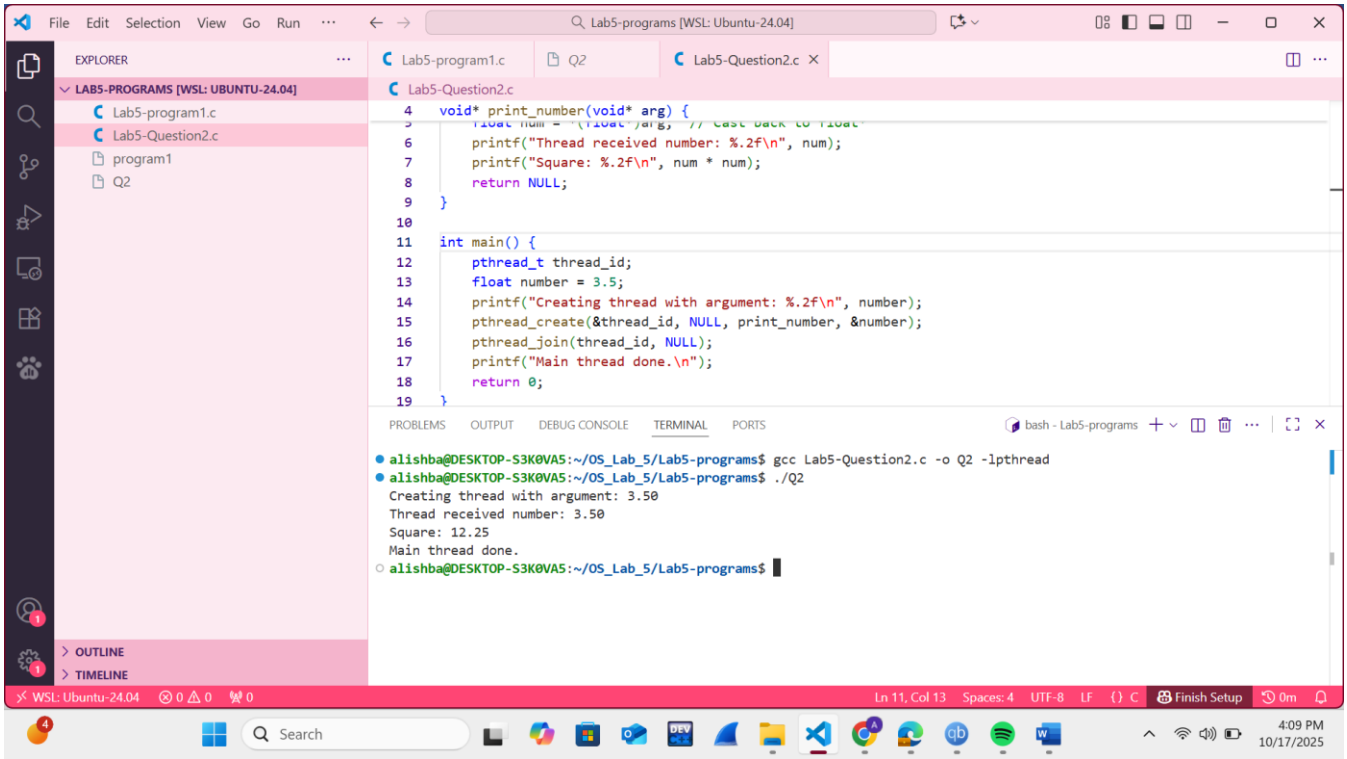
```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
void* thread_function(void* arg) {
printf("Hello from the new thread!\n");
printf("Thread ID: %lu\n", pthread_self());
return NULL;
}
int main() {
pthread_t thread_id;
printf("Main thread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


File Edit Selection View Go Run
lab_5 [WSL: Ubuntu-24.04]
EXPLORER
LAB_5 [WSL: UBUNTU-24.04]
Welcome
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Generate New Workspace...
Walkthroughs
Get started with VS Code
Customize your editor, learn the basics, and start coding
Get Started with WSL New
Learn the Fundamentals
Recent
Lab5-programs [WSL: Ubuntu-24.04] ~/OS_Lab_5
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
alishba@DESKTOP-S3K0VA5:~/lab_5$ cd ~/OS_Lab_5
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5$ mkdir Lab5-programs
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5$ cd Lab5-programs
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$ touch Lab5-program1.c
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$ code .
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$
OUTLINE
TIMELINE
WSL: Ubuntu-24.04
Finish Setup
Search
3:55 PM
10/17/2025


File Edit Selection View Go Run
Lab5-programs [WSL: Ubuntu-24.04]
EXPLORER
LAB5-PROGRAMS [WSL: UBUNTU-24.04]
Lab5-program1.c
Lab5-Question2.c
program1
Q2
Lab5-program1.c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
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);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Lab5-programs
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$ gcc Lab5-program1.c -o program1 -lpthread
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$ ./program1
Main thread starting...
Main Thread ID: 127455510910784
Hello from the new thread!
Thread ID: 127455507510976
Main thread exiting...
alishba@DESKTOP-S3K0VA5:~/OS_Lab_5/Lab5-programs$
OUTLINE
TIMELINE
WSL: Ubuntu-24.04
Ln 20, Col 1 Spaces: 4 UTF-8 LF C
Finish Setup
Search
4:07 PM
10/17/2025

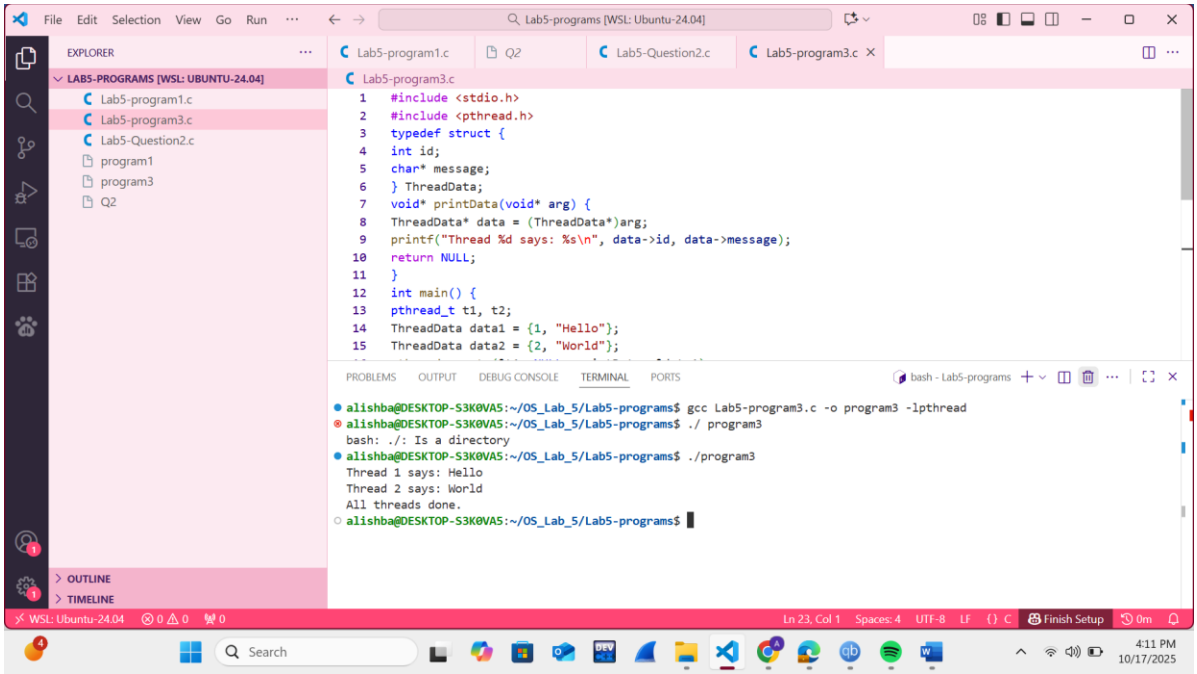
Question No.2:

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

void* print_number(void* arg) {
    float num = *(float*)arg;  // Cast back to float*
    printf("Thread received number: %.2f\n", num);
    printf("Square: %.2f\n", num * num);
    return NULL;
}

int main() {
    pthread_t thread_id;
    float number = 3.5;
    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;
}
```



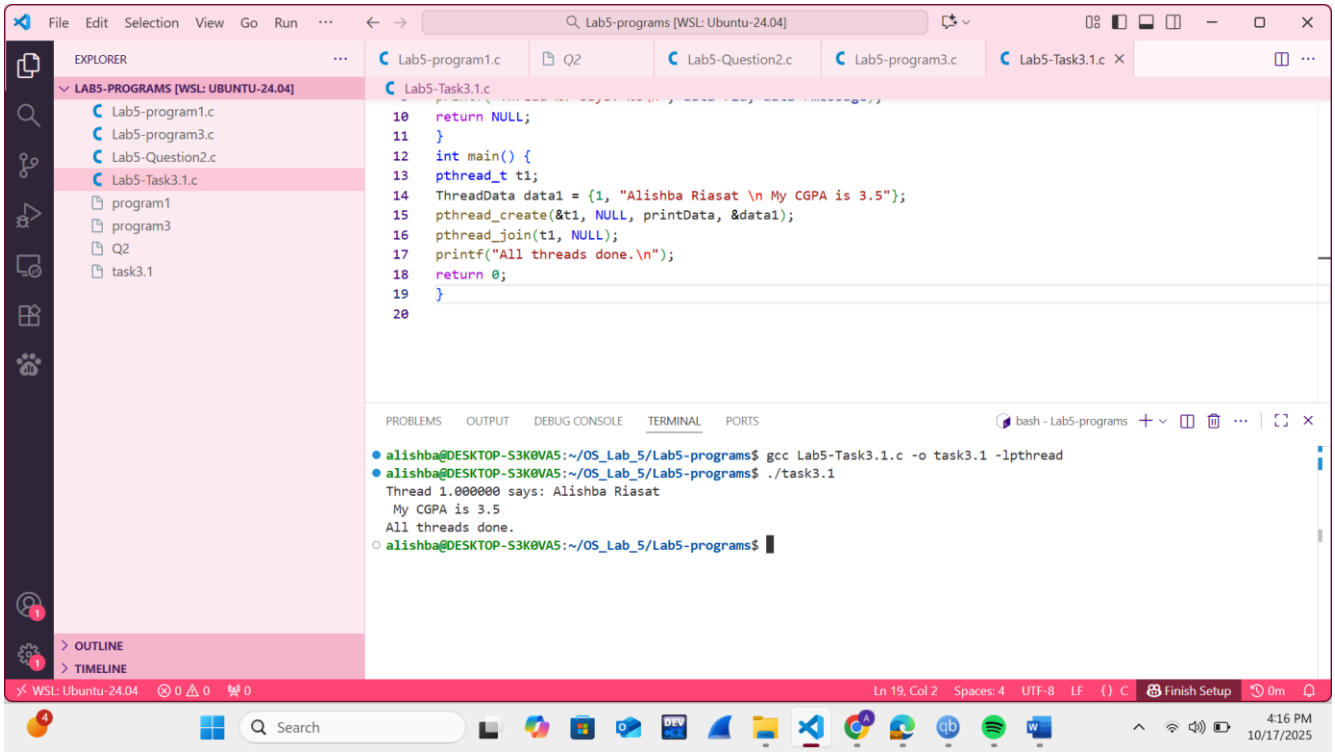
Program 3:

```c
#include <stdio.h>
#include <pthread.h>
typedef struct {
int id;
char* message;
} ThreadData;
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;
}
```



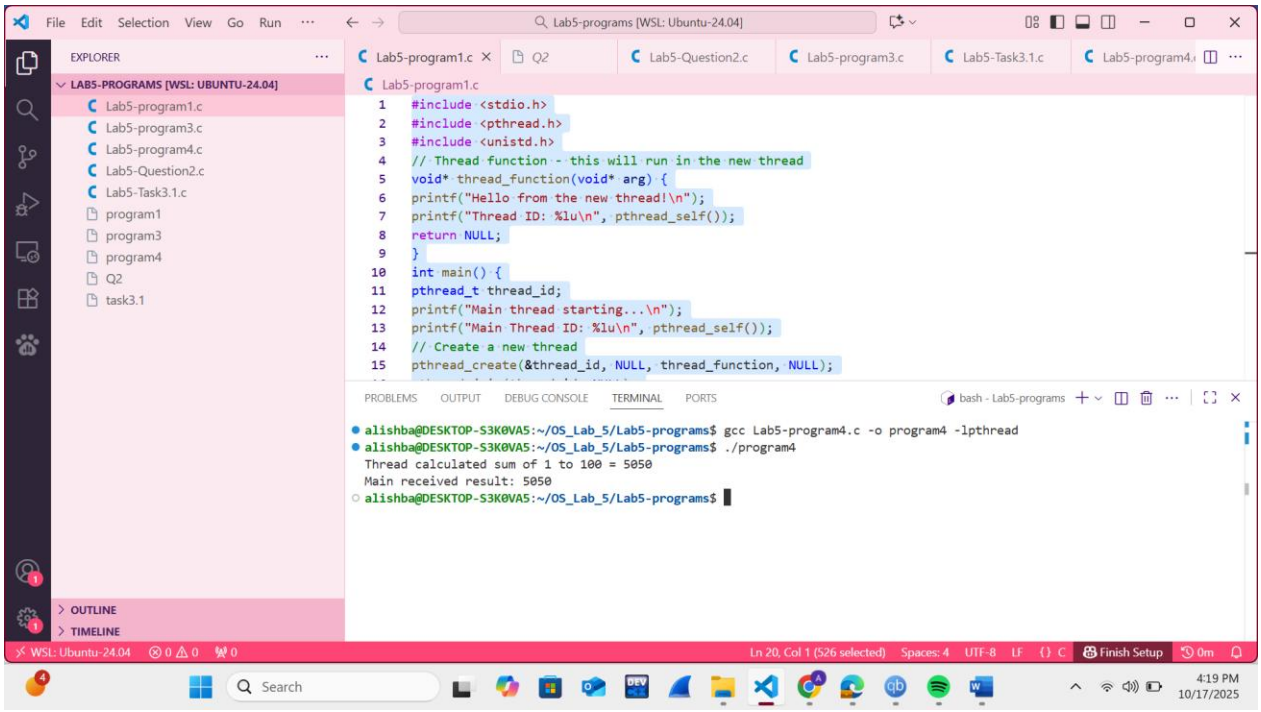
Task 3.1 (Name & Cgpa)

```c
#include <stdio.h>
#include <pthread.h>
typedef struct {
float id;
char* message;
} ThreadData;
void* printData(void* arg) {
ThreadData* data = (ThreadData*)arg;
printf("Thread %f says: %s\n", data->id, data->message);
return NULL;
}
int main() {
pthread_t t1;
ThreadData data1 = {1, "Alishba Riasat \n My CGPA is 3.5"};
pthread_create(&t1, NULL, printData, &data1);
pthread_join(t1, NULL);
printf("All threads done.\n");
return 0;
}
```



Program 4:

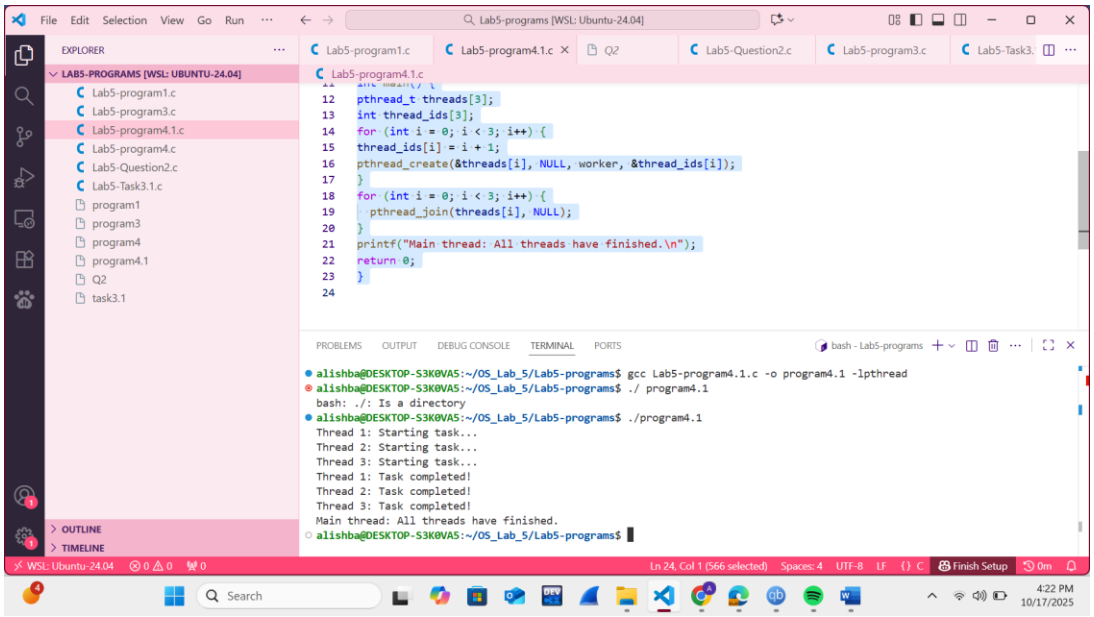
```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
void* thread_function(void* arg) {
printf("Hello from the new thread!\n");
printf("Thread ID: %lu\n", pthread_self());
return NULL;
}
int main() {
pthread_t thread_id;
printf("Main thread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



4. Basic Multithreading

Program1:

```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;
}
```



Program 2:

```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;
}
```