

Name: K.SaiKrishna

Reg-No: 192311106

30. Write C programs to demonstrate the following thread related concepts.

Aim: To demonstrate creating and terminating threads using `pthread` library.

Algorithm:

1. Include the necessary libraries.
2. Define a thread function.
3. Create threads using `pthread_create`.
4. Wait for the thread to complete using `pthread_join`.
5. Terminate the thread.

Code:

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

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

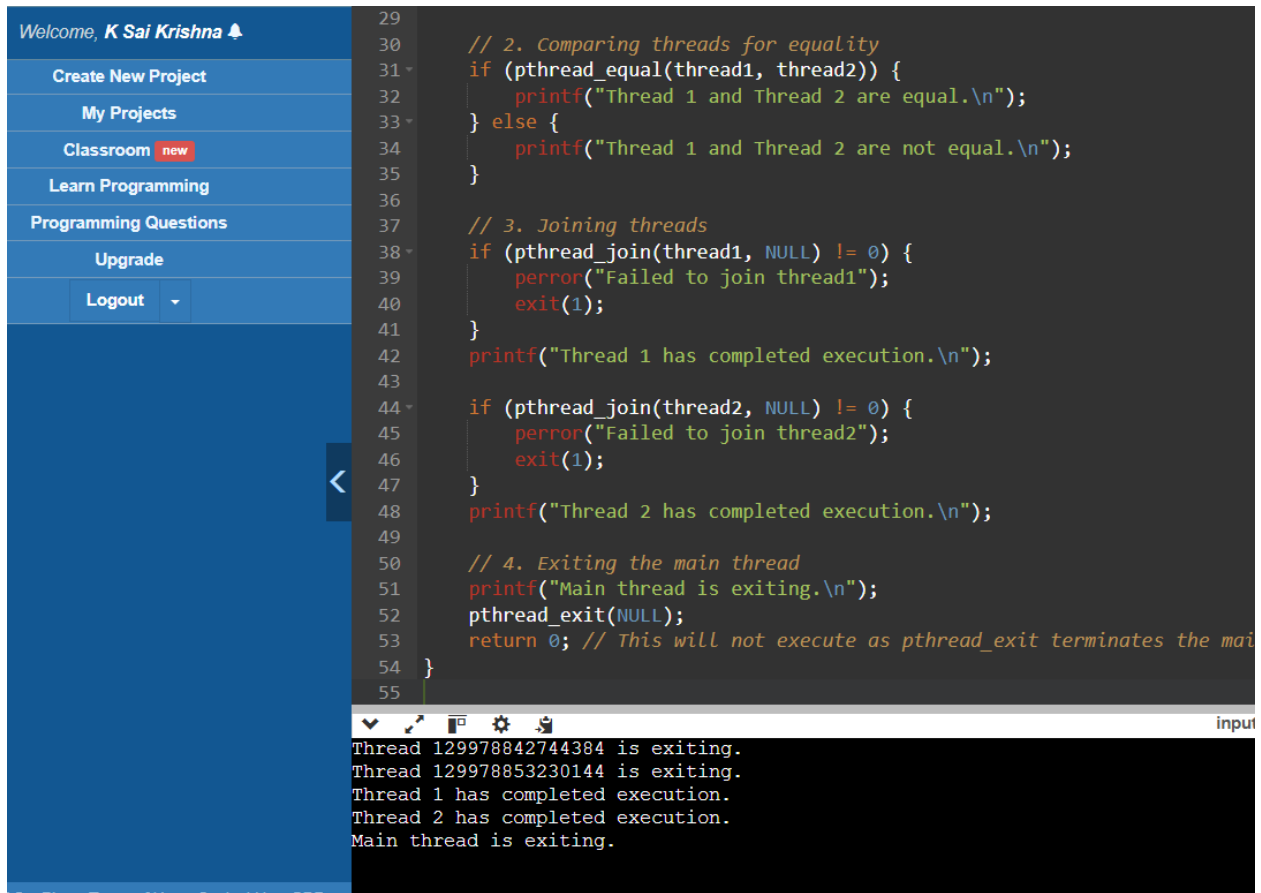
```
#include <stdlib.h>
```

```
void* thread_function(void* arg) {  
  
    printf("Thread running...\n");  
  
    return NULL;  
  
}
```

```
int main() {  
  
    pthread_t thread;  
  
    pthread_create(&thread, NULL, thread_function, NULL);  
  
    pthread_join(thread, NULL);  
  
    printf("Thread terminated.\n");  
  
    return 0;  
}
```

}

Output:



The screenshot displays a web-based IDE interface. On the left is a sidebar with a user profile 'Welcome, K Sai Krishna' and navigation links: 'Create New Project', 'My Projects', 'Classroom' (marked 'new'), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area shows a C program using pthreads. The code includes comments for each section: '2. Comparing threads for equality', '3. Joining threads', and '4. Exiting the main thread'. It creates two threads, compares them, joins them, and then exits. The output window at the bottom shows the execution results: 'Thread 129978842744384 is exiting.', 'Thread 129978853230144 is exiting.', 'Thread 1 has completed execution.', 'Thread 2 has completed execution.', and 'Main thread is exiting.'

```
29
30 // 2. Comparing threads for equality
31 if (pthread_equal(thread1, thread2)) {
32     printf("Thread 1 and Thread 2 are equal.\n");
33 } else {
34     printf("Thread 1 and Thread 2 are not equal.\n");
35 }
36
37 // 3. Joining threads
38 if (pthread_join(thread1, NULL) != 0) {
39     perror("Failed to join thread1");
40     exit(1);
41 }
42 printf("Thread 1 has completed execution.\n");
43
44 if (pthread_join(thread2, NULL) != 0) {
45     perror("Failed to join thread2");
46     exit(1);
47 }
48 printf("Thread 2 has completed execution.\n");
49
50 // 4. Exiting the main thread
51 printf("Main thread is exiting.\n");
52 pthread_exit(NULL);
53 return 0; // This will not execute as pthread_exit terminates the main
54 }
55
```

Thread 129978842744384 is exiting.
Thread 129978853230144 is exiting.
Thread 1 has completed execution.
Thread 2 has completed execution.
Main thread is exiting.

RESULT

This experiment successfully demonstrates basic thread operations in C using the pthread library, including thread creation, synchronization (joining), comparison, and proper exit handling.

