Multithreading Online Proctoring System

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

// ----------------------------
// THREAD 1: Video Recording
// ----------------------------
void* videoRecorder(void* arg) {
    for (int i = 1; i <= 5; i++) {
        printf("[Video Thread] Captured frame %d\n", i);
        sleep(1); // simulate time delay
    }
    return NULL;
}

// ----------------------------
// THREAD 2: Mouse Monitoring
// ----------------------------
void* mouseMonitor(void* arg) {
    for (int i = 1; i <= 5; i++) {
        int x = rand() % 800;
        int y = rand() % 600;
        printf("[Mouse Thread] Mouse at (%d, %d)\n", x, y);
        usleep(500000); // 0.5 seconds
    }
    return NULL;
}

// ----------------------------
// THREAD 3: Tab Switching Detection
// ----------------------------
void* tabDetector(void* arg) {
    char *tabs[] = {"Exam Portal", "YouTube", "WhatsApp Web", "Google", "Mail"};

    for (int i = 1; i <= 5; i++) {
        int t = rand() % 5;
        printf("[Tab Thread] Active tab: %s\n", tabs[t]);
        usleep(700000); // 0.7 seconds
    }
    return NULL;
}
```

```c
int main() {
    pthread_t t1, t2, t3;

    printf("\n--- Online Proctoring System Started ---\n\n");

    // Create three threads
    pthread_create(&t1, NULL, videoRecorder, NULL);
    pthread_create(&t2, NULL, mouseMonitor, NULL);
    pthread_create(&t3, NULL, tabDetector, NULL);

    // Wait for all threads to finish
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);

    printf("\n--- Monitoring Completed ---\n");

    return 0;
}
```