```
import java.util.concurrent.*;
import java.util.ArrayList;

public class Thread_join {
	/*
	 * ArrayBlockingQueue is a synchronized queue structure
	 */
	final static int MAX = 10;
	protected ArrayBlockingQueue<Integer> q = new ArrayBlockingQueue<Integer>(MAX);
	protected boolean done = false;		//Flag for main to signify a stop
	static ArrayList<Thread> tname = new ArrayList<Thread>();

	public Thread_join(int NumT1, int NumT2) {

		for (int i = 0; i < NumT1; i++) {
			tname.add(new Thread(new T1()));
		}
		for (int i = 0; i < NumT2; i++) {
			tname.add(new Thread(new T2()));
		}
		for (int i=0; i<tname.size(); i++) {
			tname.get(i).start();
		}

	}

	public class T1 implements Runnable {
		public void run() {
			int i = 0;
			while (!done) {
				try {
					Thread.sleep(1000);
				} catch (InterruptedException ex) {
					return;
				}
			if (q.offer(i) == true) {
				System.out.println("T1 adding " + i );
				i++;
			} else {
				System.out.println("Unable to add");
			}
		}
		}
	}

	public class T2 implements Runnable {
		public void run() {
			int i;
			while (!done) {
				try {
					Thread.sleep(1000);
				} catch (InterruptedException ex) {
					return;
				}
				try {
					i = q.take();
					System.out.println("T2 taking " + i);
				} catch (InterruptedException e) {
					return;
				}
			}
		}
	}

	public static void main(String[] args) {

		Thread_join pct = new  Thread_join(1,5);
		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			e.printStackTrace();
		}
		pct.done = true;
		for (int i=0; i<tname.size(); i++) {
			tname.get(i).interrupt();
			try {
				tname.get(i).join();
			} catch (InterruptedException e) {
				e.printStackTrace();
			}
		}
	}

}


T1 adding 0
T2 taking 0
T1 adding 1
T2 taking 1
T1 adding 2
T2 taking 2
T1 adding 3
T2 taking 3
```

CS 370 Worksheet #10 Name_____

1) What is a race condition? How does it affect multithreaded applications? What are the 3 conditions necessary to solve the critical section problem?

2) How does the test and set instruction work? What property must it have for it to work correctly?

3) What is the difference between a binary and a counting semaphore? Can a binary semaphore be implemented as a counting semaphore? Can a counting semaphore be implemented as a counting semaphore?

4) What is deadlock?