Multithreading Communication Exercise:

Part I

Write a program in C or Java which implements the following:

1. The program will start and create a table of N integer where N is given as input by the user, for instance array [N].
2. Then the program will fill the table with N integer numbers, randomly selected from the [-100, 100] field of numbers.
3. Then the program will create 4 threads, each of which will do the following:

   i. The first thread will calculate the sum T1 of N elements.

   ii. The second thread will calculate the product T2 of N elements.

   iii. The third thread will select a random number from [-1000,1000] field.
   i.e. T3 = random (X): -1000 <= X <= 1000

   iv. The forth thread will "sort" the three previous threads in a row from 1 to 3, depending on their results as T1, T2, T3.

For example: If T1 = 1200, T2 = -53521 and T3 = 850, then the order is: T1, T3, T2.

Part II

Based on the program you implemented above, write a modified version, which will additionally implement the following:

1. When each of the first two threads executes, it will randomly select the half of the table, and will change it with new random numbers from the [-100, 100] field. Then the program will count, as before, the sum (first thread) or the product (second thread) of the modified table.

2. In this case, you should use a mutual blocking threads method (i.e. mutex / semaphores, sleep / weakup calls etc) so there is never a chance to modify both threads at the same time some shared table elements.

In each program, indicative messages should be displayed each time running a thread, waiting for another, completing his work