Multithreading Excercise

1. Write a Java program to perform a runnable interface, take two threads t1 and t2 and fetch the names of the thread using getName() method.

   Expected output:
   Thread names are following:
   Thread A
   Thread B

2. You need to run this following simple program with at least three threads (name those threads as A,B,C....), and find the output.
   i. Initialize a variable sum = 0
   ii. Run a for loop for 10 times. i.e : for(int x = 0; x < 10; x++) . Add the value x to the value sum in each iteration
   iii. Print the thread name, and the value as follows : "Thread : A - value : 45 " in each iteration
   iv. Finally Print the sum from each thread as follows : "Thread : A - Sum : 45 "
3. Write a program to print "Good morning" and "Welcome" continuously on the screen in Java using threads.
4. Add a step method in the welcome thread of question 3 to delay its execution for 200ms.
5. Which integer between 1 and 10000 has the largest number of divisors, and how many divisors does it have? Write a program to find the answers and print out the results. It is possible that several integers in this range have the same, maximum number of divisors. Your program only has to print out one of them.

   Now write a program that uses multiple threads to solve the same problem, but for the range 1 to 100000. By using threads, your program will take less time to do the computation when it is run on a multiprocessor computer. At the end of the program, output the elapsed time, the integer that has the largest number of divisors, and the number of divisors that it has. For this exercise, you should simply divide up the problem into parts and create one thread to do each part.