-- Student Vs Question assignment

Aayush Aayush <aayush@accolitedigital.com> -- Q.6

Adishi Sood <adishi@accolitedigital.com> -- Q.2

Aparna Kochar <aparna.kochar@accolitedigital.com> -- Q.1

Himanshu Kumar <himanshu.kumar@accolitedigital.com> -- Q.3

Hitesh Kumar <hitesh.kumar@accolitedigital.com> -- Q.5

Mohit Sharma <mohit.sharma1@accolitedigital.com> -- Q.4

Palkin Khurana <palkin.khurana@accolitedigital.com> -- Q.4

Paras Tyagi <paras.tyagi@accolitedigital.com> -- Q.6

Priyanshu Bhardwaj <priyanshu.bhardwaj@accolitedigital.com> -- Q.2

Rakti Singal <rakti.singal@accolitedigital.com> -- Q.6

Satyam Mishra <satyam.mishra@accolitedigital.com> -- Q.5

Simran Srivastava <simran.srivastava1@accolitedigital.com> -- Q.1

Vaibhav Chotani <vaibhav.chotani@accolitedigital.com> -- Q.6

Vikram Ohri <vikram.ohri@accolitedigital.com> -- Q.3

1. Create a custom BlockinStack.

- which blocks the consumer threads from popping elements until the stack is not empty.

- Multiple threads should be able to do push(Producers) & pop(Consumers) at the same time.

- Do Not use anything from the concurrent package.

2. Print Fibonacci series.

- Each number in the series should be printed by a unique thread.

- Use, ExecutorService, Callable & Future

- After the series is printed, a report should be printed displaying which thread(name) printed which number.

3. Demonstrate DeadLock.

- Use, ExecutorService, Callable & Future

4. Print Factorial series.

- Each number in the series should be printed by a unique thread.

- Do not use anything from the concurrent package.

- After the series is printed, a report should be printed displaying which thread(name) printed which number.

5. Three threads synchronization.

- Thread-1 is incrementing a counter(c1) value by 1 'n' times.

- When ever counter value is multiple of 25, then Thread-2 should increment another counter say c2 by 2;

- Whenever c2 is a multiple of 4 then another thread T3 should update counter c3 by 1;

- Finally print all the counters.

- Only input to the program should be 'n'.

- DO NOT Use anything from concurrent package

6. Horse race.

- There are 'N' horses, denoted by 'N' threads.

- The speed of each horse changes to a random number from 40-60 KM/H, every 30 seconds.

- Should use a seperate thread to generate and set speed randomly.

- The race is for 5 minutes.

- Race track is 10 KM.

- Once all horses finish the race, the program should announce the winner.

- Program should also print statistics like speed history, average speed, top speed & low speed for each horse.

- Statistic calculation should happen in parallel using a separate thread.

- Use any java lib to achieve this.

- Only input to the program is 'N'