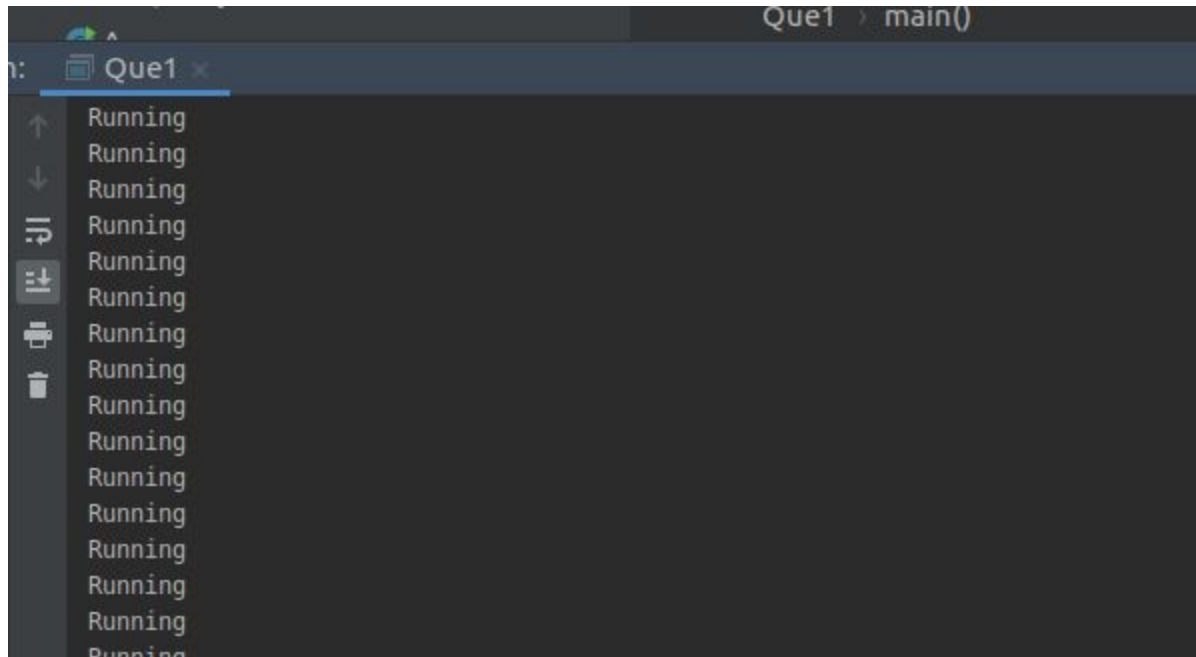
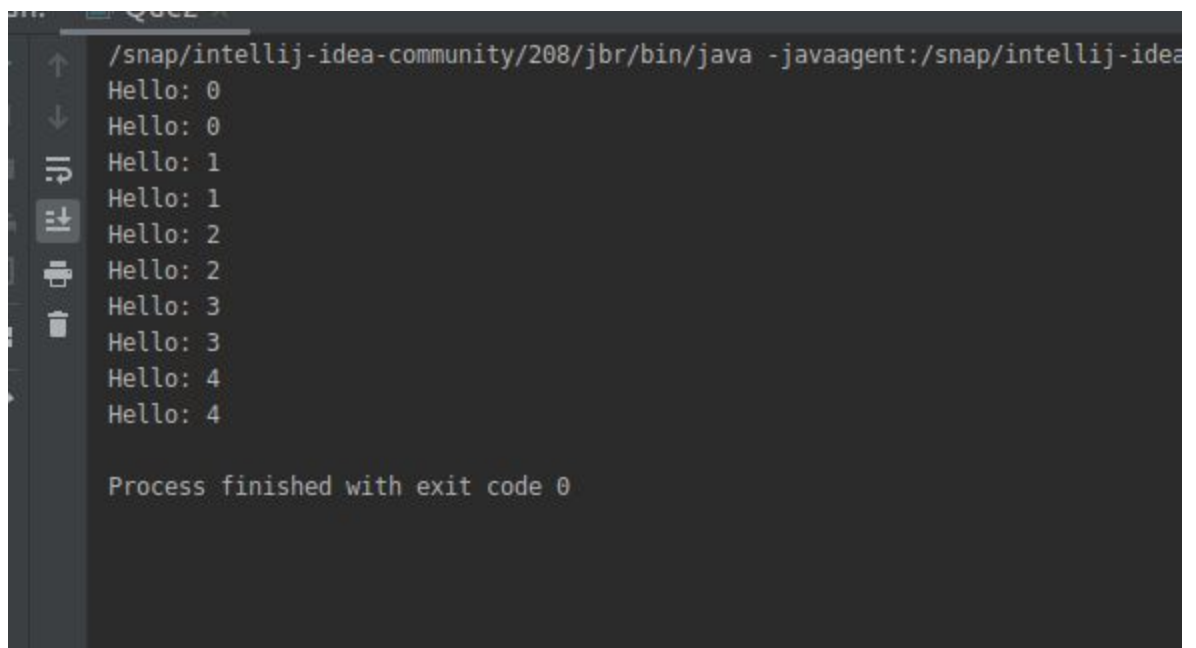


## Basics of Multi Threading

1. Write a program do to demonstrate the use of volatile keyword.

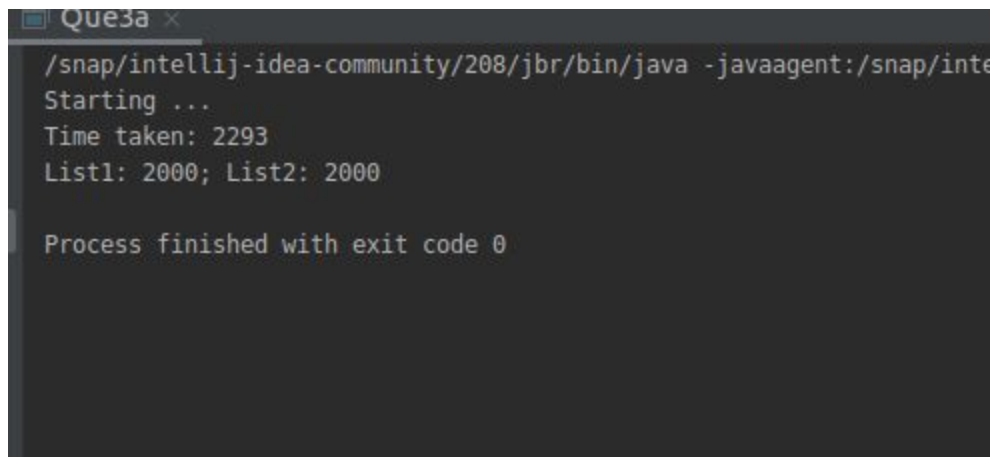


2. Write a program to create a thread using Thread class and Runnable interface each.



3. Write a program using synchronization block and synchronization method

ANS> synchronization block



```
Que3a x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/jbr/bin/javaagent.jar
Starting ...
Time taken: 2293
List1: 2000; List2: 2000

Process finished with exit code 0
```

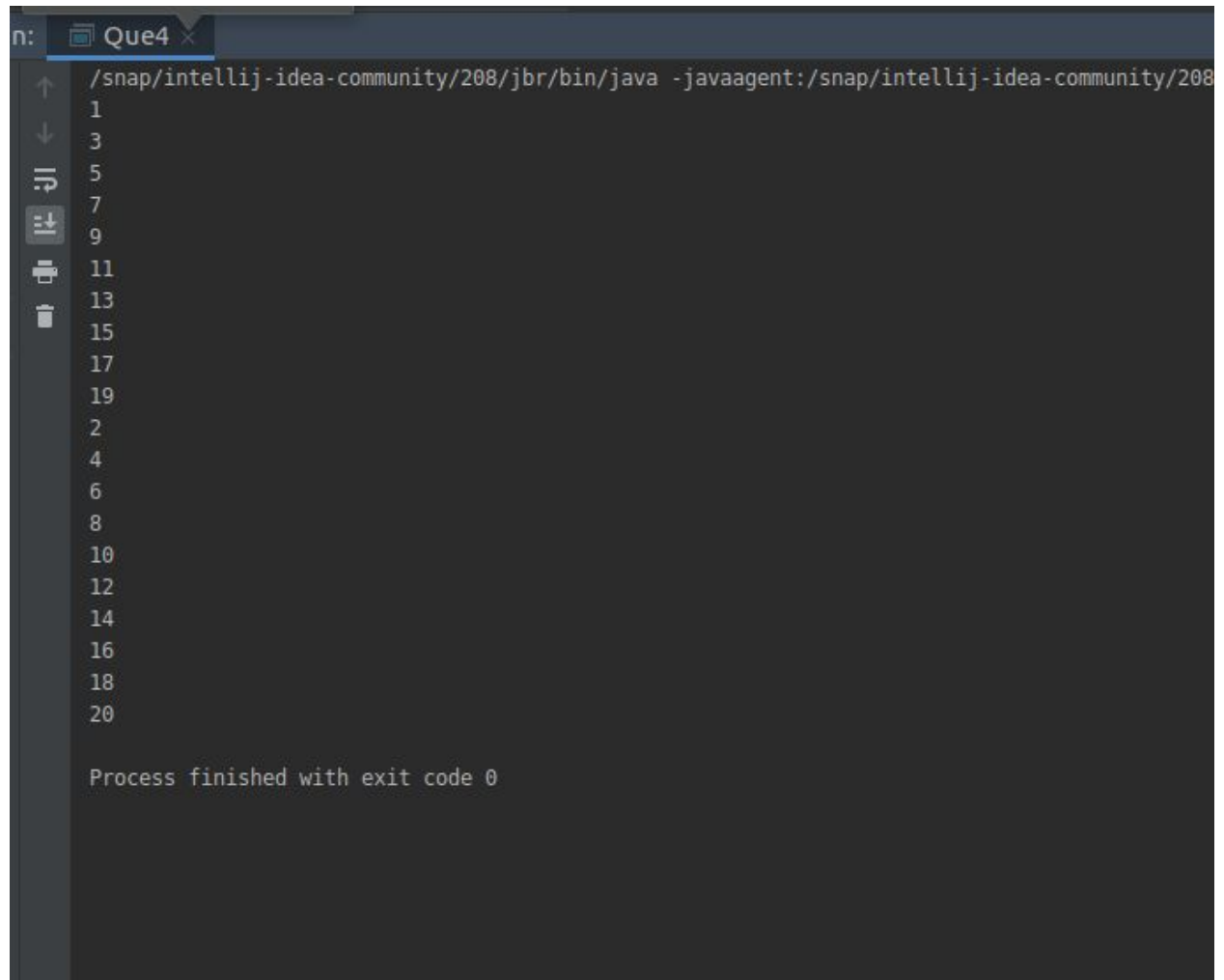
synchronization method



```
Que3b x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/jbr/bin/javaagent.jar
Count is: 20000

Process finished with exit code 0
```

4. Write a program to create a Thread pool of 2 threads where one Thread will print even numbers and other will print odd numbers.



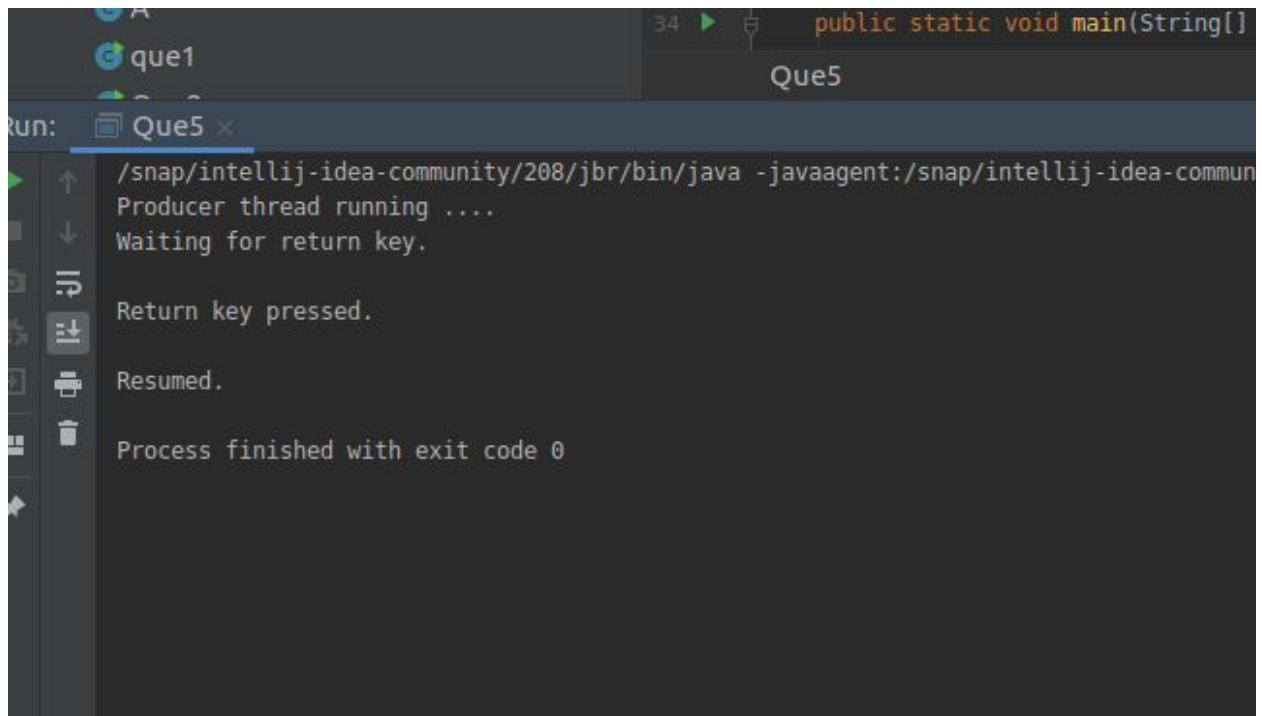
The screenshot shows a terminal window titled "Que4" with the following output:

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208
1
3
5
7
9
11
13
15
17
19
2
4
6
8
10
12
14
16
18
20

Process finished with exit code 0
```

The output displays odd numbers from 1 to 19 in the first sequence and even numbers from 2 to 20 in the second sequence, demonstrating the execution of two parallel threads.

5. Write a program to demonstrate wait and notify methods.



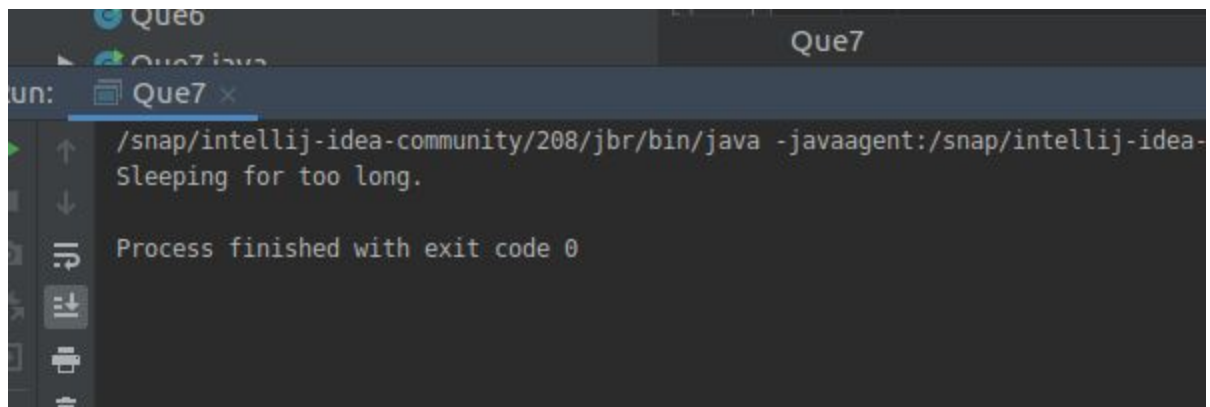
```
public static void main(String[] args) {  
    // ...  
    Producer thread running ....  
    Waiting for return key.  
    Return key pressed.  
    Resumed.  
    Process finished with exit code 0  
}
```

6. Write a program to demonstrate sleep and join methods.



```
// ...  
thread 2 is called  
thread 1 is called  
finally main block reach to end  
Process finished with exit code 0
```

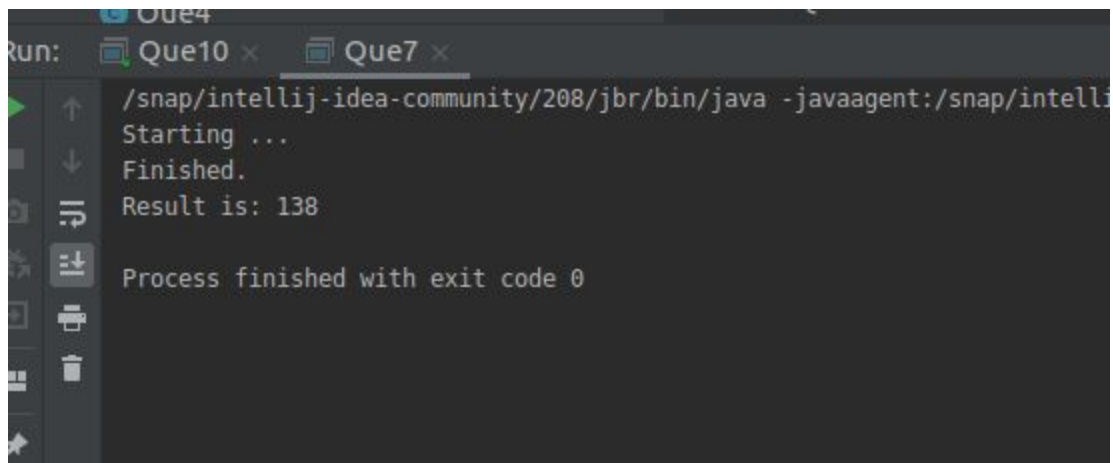
7. Run a task with the help of callable and store its result in the Future.



The screenshot shows the IntelliJ IDEA Run console with a single tab labeled 'Que7'. The console output is as follows:

```
Run: Que7 x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-
Sleeping for too long.

Process finished with exit code 0
```

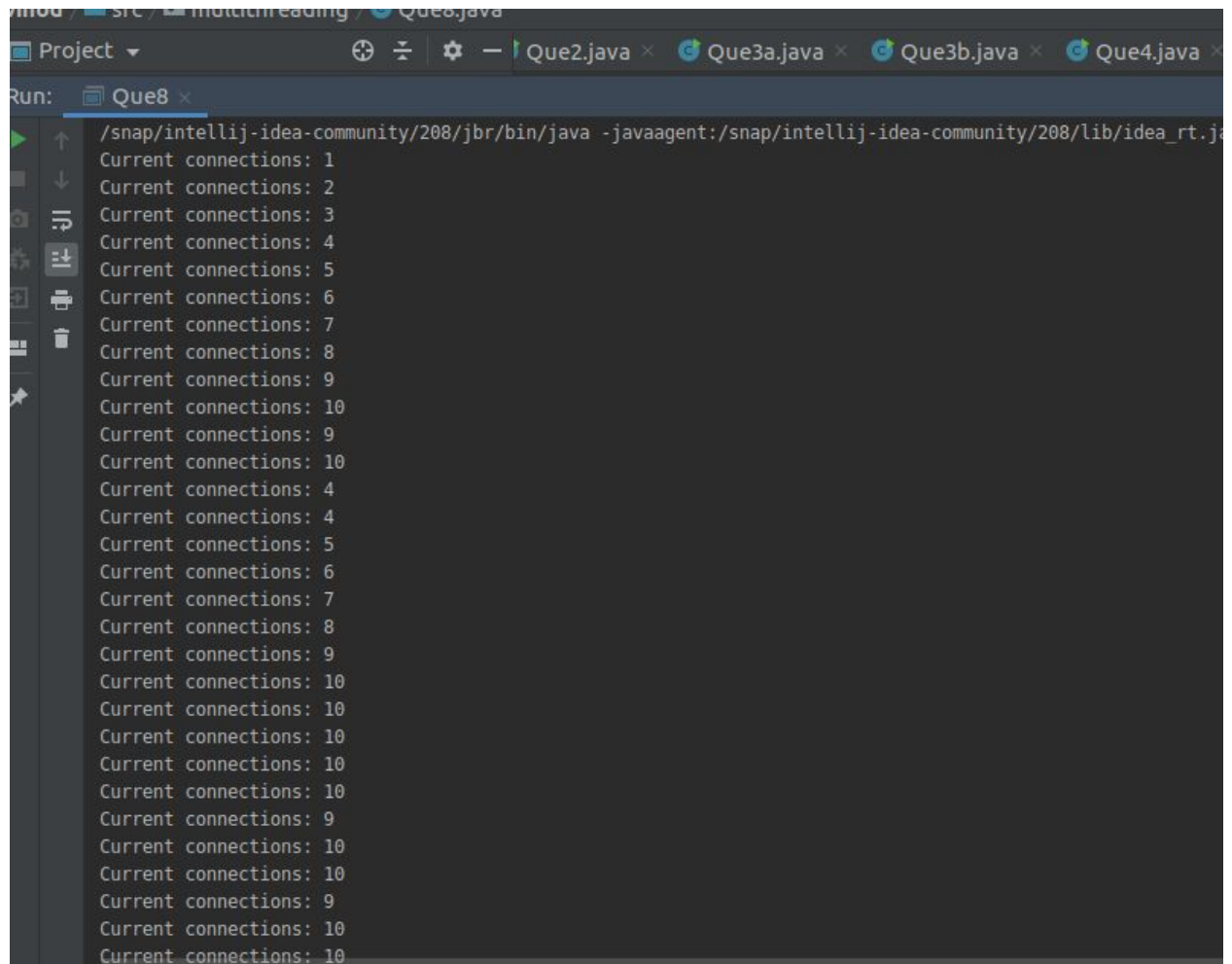


The screenshot shows the IntelliJ IDEA Run console with two tabs: 'Que10' and 'Que7'. The 'Que7' tab is active, and the console output is as follows:

```
Run: Que10 x Que7 x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intelli
Starting ...
Finished.
Result is: 138

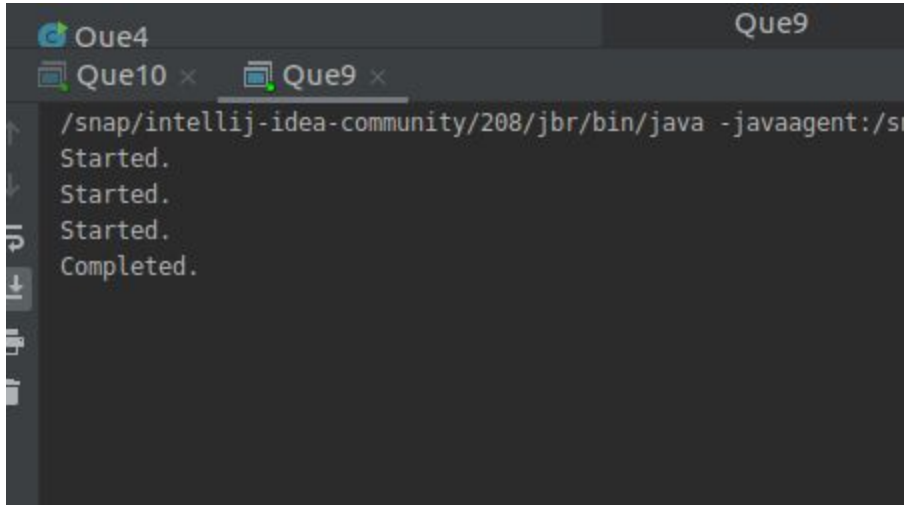
Process finished with exit code 0
```

8. Write a program to demonstrate the use of semaphore

A screenshot of an IDE window showing a Java program's output. The window has a title bar with 'Project' and several tabs for 'Que2.java', 'Que3a.java', 'Que3b.java', and 'Que4.java'. The active tab is 'Que8'. The output area shows a sequence of 'Current connections: X' messages, where X ranges from 1 to 10. The sequence is: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 10, 4, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 10, 10, 9, 10, 10, 9, 10, 10. The IDE interface includes a 'Run' button and a toolbar with various icons for file operations and debugging.

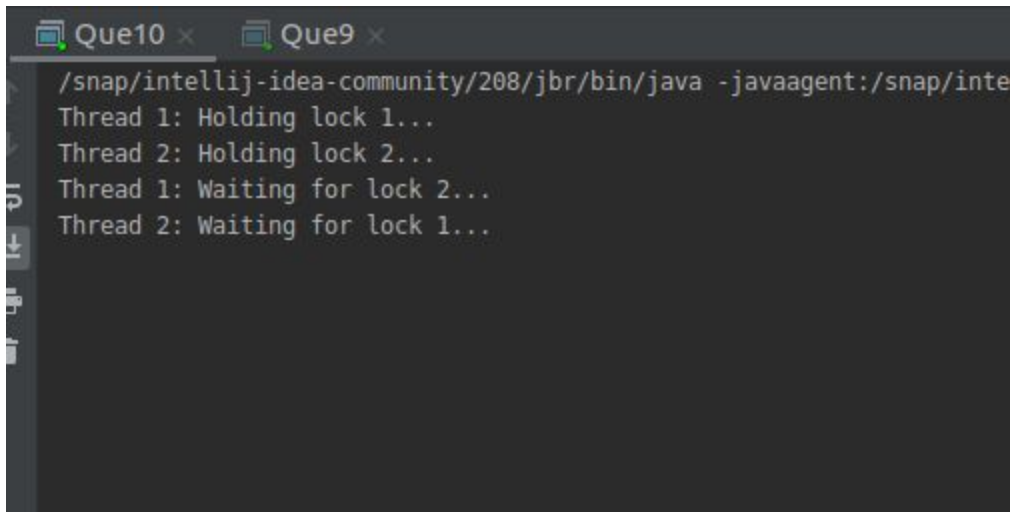
```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar
Current connections: 1
Current connections: 2
Current connections: 3
Current connections: 4
Current connections: 5
Current connections: 6
Current connections: 7
Current connections: 8
Current connections: 9
Current connections: 10
Current connections: 9
Current connections: 10
Current connections: 4
Current connections: 4
Current connections: 5
Current connections: 6
Current connections: 7
Current connections: 8
Current connections: 9
Current connections: 10
Current connections: 10
Current connections: 10
Current connections: 10
Current connections: 10
Current connections: 9
Current connections: 10
Current connections: 10
Current connections: 9
Current connections: 10
Current connections: 10
```

9. Write a program to demonstrate the use of CountDownLatch



```
Que4
Que10 x Que9 x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/sn
Started.
Started.
Started.
Completed.
```

10. Write a program which creates deadlock between 2 threads



```
Que10 x Que9 x
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/inte
Thread 1: Holding lock 1...
Thread 2: Holding lock 2...
Thread 1: Waiting for lock 2...
Thread 2: Waiting for lock 1...
```