Worksheet 15a: Data Driven Futures (turn page over for worksheet 15b)

Name: ______________________ Netid: ______________________

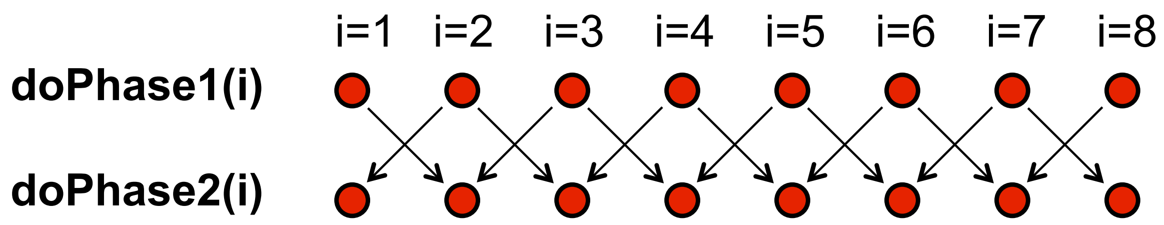
For the example below, will reordering the five async statements change the meaning of the program (assuming that the semantics of the reader/writer methods depends only on their parameters)? If so, show two orderings that exhibit different behaviors. If not, explain why not.

```
DataDrivenFuture left = new DataDrivenFuture();
DataDrivenFuture right = new DataDrivenFuture();
finish {
  async await(left) leftReader(left); // Task3
  async await(right) rightReader(right); // Task5
  async await(left,right)
        bothReader(left,right); // Task4
  async left.put(leftWriter()); // Task1
  async right.put(rightWriter());// Task2
}
```

Worksheet #15b: Left-Right Neighbor Synchronization using Phasers (turn page over for worksheet 15a)

Name: ______________________

Netid: ______________________



Complete the phased clause below to implement the point-to-point synchronization shown above for m tasks (generalization of slide 11)

```
finish (() -> {
  final HjPhaser[] ph =
      new HjPhaser[m+2]; // array of phaser objects
  forseq(0, m+1, (i) -> { ph[i] = newPhaser(SIG_WAIT) });
  forseq(1, m, (i) -> {
    asyncPhased(
       ph[i-1].inMode(......),
       ph[i].inMode(......),
       ph[i+1].inMode(......), ()->{
     doPhase1(i);
     next();
     doPhase2(i); }); // asyncPhased
  }); // forseq
}); // finish
```