**Q4 Process Synchronization - Semaphores (20 points)**

Let 9 processes, p1 to p9, have the following precedence constraints. Initially, only p1 and p2 start. p3 and p4 start only after both p1 and p2 finish. p5, p6, and p7 start only when p4 finishes. p8 starts only after p3 and p5 finish. The last process, p9, starts only when p6, p7, and p8 finish. Show how these constraints can be implemented using semaphores. Appropriate wait and signal operations should be inserted at the beginning and end of the code for each process.

**Answer**

```
semaphore P1_done = 0;
semaphore P2_done = 0;
semaphore P3_done = 0;
semaphore P4_done = 0;
semaphore P5_done = 0;
semaphore P6_done = 0;
semaphore P7_done = 0;
semaphore P8_done = 0;

P1: <execute>
signal(P1_done);    // Let go off P3
signal(P1_done);    // Let go off P4

P2: <execute>
signal(P2_done);    // Let go off P3
signal(P2_done);    // Let go off P4

P3: wait(P1_done);
wait(P2_done);
<execute>
signal(P3_done);    // Let go off P8

P4: wait(P1_done);
wait(P2_done);
<execute>
signal(P4_done);    // Let go off P5
signal(P4_done);    // Let go off P6
signal(P4_done);    // Let go off P7

P5: wait(P4_done);
<execute>
signal(P5_done);    // Let go off P8

P6: wait(P4_done);
<execute>
signal(P6_done);    // Let go off P9

P7: wait(P4_done);
<execute>
signal(P7_done);    // Let go off P9

P8: wait(P3_done);
wait(P5_done);
<execute>
signal(P8_done);    // Let go off P9

P9: wait(P6_done);
wait(P7_done);
wait(P8_done);
<execute>
```

**Q5 Monitors (25 points)**

A file is to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: The sum of all unique numbers associated with all the processes currently accessing the file must be less than n. Write a monitor to coordinate access to the file.

ANSWER:

```
monitor fileaccess {
  int currsum = 0;
  int n;
  condition c;

  void accessfile(int mynum) {
       while (currsum + mynum >= n)
     c.wait();
     currsum += mynum;
  }

  void finishaccess(int mynum) {
     currsum -= mynum;
     c.signal();
  }
}
```