1. Suppose the following two processes, `foo` and `bar` are executed concurrently and share the semaphore variables `S` and `R` (each initialized to 1) and the integer variable `x` (initialized to 0).

   ```c
   void foo( ) {
     do {
       S.wait( );
       R.wait( );
       x++;
       S.signal( );
       R.signal( );
     } while (1);
   }

   void bar( ) {
     do {
       R.wait( );
       S.wait( );
       x--;
       S.signal( );
       R.signal( );
     } while (1);
   }
   ```

   (a) Can the concurrent execution of these two processes result in one or both being blocked forever? If yes, give an execution sequence in which one or both are blocked forever.

   (b) Can the concurrent execution of these two processes result in the indefinite postponement of one of them? If yes, give an execution sequence in which one is indefinitely postponed.
2. Suppose that 100 instances of the following process, widget, are executed concurrently, sharing the semaphore variable F which is initialized to 7.

```c
#include <stdio.h>
void main( ) {
    FILE *ifile;
    int  x  = 25;
    char ch = ' ';
    ifile = open("widget.c", "r");
    do {
        F.wait( );
        x--;
        ch = fgetc(ifile);
        F.signal( );
    } while (x);
    fclose(ifile);
}
```

(a) What is the effect of the semaphore F? Be very specific.

(b) Assume the 100 instances of widget are created simultaneously on a single-processor machine, and that one instance of widget running on the same processor would require M seconds to complete. Assume also that system overhead (for process switching, etc.) is zero.

(i) What will be the turnaround time for the first instance of widget that completes?

(ii) What will be the turnaround time for the last instance of widget that completes?