Incorrect solution for producer/consumer problem with infinite buffer using binary semaphores.

```
/*  program producerconsumer */
      int n;
      binary_semaphore s = 1, delay = 0;
      void producer()
      {
           while (true) {
                 produce();
                 semWaitB(s);
                 append();
                 n++;
                 if (n==1) semSignalB(delay);
                 semSignalB(s);
           }
      }
      void consumer()
      {
           semWaitB(delay);
           while (true) {
                 semWaitB(s);
                 take();
                 n--;
                 semSignalB(s);
                 consume();
                 if (n==0) semWaitB(delay);
           }
      }
      void main()
      {
           n = 0;
           parbegin (producer, consumer);
      }
```

Figure 5.12 An Incorrect Solution to the Infinite-Buffer Producer/Consumer Problem Using Binary Semaphores

The correct solution with binary semaphores:

```
/* program producerconsumer */
      int n;
      binary_semaphore s = 1, delay = 0;
      void producer()
      {
           while (true) {
                 produce();
                 semWaitB(s);
                 append();
                 n++;
                 if (n==1) semSignalB(delay);
                 semSignalB(s);
           }
      }
      void  consumer()
      {
           int  m; /* a local variable */
           semWaitB(delay);
           while  (true) {
                 semWaitB(s);
                 take();
                 n--;
                 m = n;
                 semSignalB(s);
                 consume();
                 if  (m==0) semWaitB(delay);
           }
      }
      void  main()
      {
           n = 0;
           parbegin  (producer, consumer);
      }
```

Figure 5.13 A Correct Solution to the Infinite-Buffer Producer/Consumer Problem Using Binary Semaphores

A solution with counting semaphores for producer/consumer with infinite buffer:

```
/* program producerconsumer */
      semaphore n = 0, s = 1;
      void producer()
      {
           while (true) {
                 produce();
                 semWait(s);
                 append();
                 semSignal(s);
                 semSignal(n);
           }
      }
      void consumer()
      {
           while (true) {
                 semWait(n);
                 semWait(s);
                 take();
                 semSignal(s);
                 consume();
           }
      }
      void main()
      {
                parbegin  (producer, consumer);
      }
```

Figure 5.14 **A Solution to the Infinite-Buffer Producer/Consumer Problem Using Semaphores**

A solution for producer/consumer with bounded buffer:

```
/* program boundedbuffer */
      const int sizeofbuffer = /* buffer size */;
      semaphore s = 1, n = 0, e = sizeofbuffer;
      void producer()
      {
           while (true) {
                 produce();
                 semWait(e);
                 semWait(s);
                 append();
                 semSignal(s);
                 semSignal(n);
           }
      }
      void consumer()
      {
           while  (true) {
                 semWait(n);
                 semWait(s);
                 take();
                 semSignal(s);
                 semSignal(e);
                 consume();
           }
      }
      void main()
      {
           parbegin  (producer, consumer);
      }
```

Figure 5.16 **A Solution to the Bounded-Buffer Producer/Consumer Problem Using Semaphores**