How Would You Test This?

You've been given the following C++ functions which implement FIFO queues of unsigned char ("byte"):

```
Q* create_queue();  // Creates a FIFO byte queue, returning a handle to it

void destroy_queue(Q* q);           // Destroy a previously created byte queue

void enqueue_byte(Q* q, unsigned char b);   // Adds a new byte to a queue <q>

unsigned char dequeue_byte(Q* q);       // Pops the next byte off the queue <q>
```

If the functions run out of memory, it will call this function (which you may define / override):

```
void on_out_of_memory();
```

If the caller makes an illegal request, it will call this function (which you may define / override):

```
void on_illegal_operation();
```

As it turns out, the specifications also require that the implementation

- uses no more than 2048 bytes to implement all byte queues, and
- must support 15 queues with an average of 80 or so bytes in each queue.

Please see the following link for sample output and sample implementation:

```
https://github.com/KageKirin/CodingTests/tree/master/SuckerPunch
```

Questions

1. Please describe how would you test the code which implements the above specification.
2. Please write the C++ code which implements your proposed tests.

Caution

Please do not submit your solution as a pull request against the `KageKirin` repo, as we do *not* control that repo (it contains someone's solution to Sucker Punch's (different) take-home test):

1. Do not disadvantage yourself by allowing other candidates to base their solution upon yours.
2. Please help us maintain the integrity of this take-home test.

Thank you in advance for your cooperation.