We here list some representative solutions for your reference. Please bare in mind that solutions to the pop-quiz are not limited to the following ones.

1) Introduce a counter (By 李岩松):

```
Initialize counter=0;
Producer:
While (true){
      /produce an item/
      While ((((in+1) % buffer_size)==out) && (counter==buffer_size))
            /do nothing/;
      buffer[in]=item;
      in=(in+1)%buffer_size;
      counter++;
}

Consumer:
While (true){
      while ((in==out) && (counter==0))
            /do nothing/
       /remove/
       item=buffer[out];
       out=(out+1)%buffer_size;
       counter--;
       return item;
}
```

2) Introduce a flag (By 武晨阳):

Introduce a new variable isFull (Boolean)

Producer:

```
while  (true)
{
   /*  Produce  an  item  */
```

```
    while (in==out && isFull==true)
       ; /* do nothing -- no free buffers */
    buffer[in] = item;
    in = (in + 1) % BUFFER_SIZE;
    if(in==out) isFull = true;
}
```

Customer:

```
while (true) {
    while (in == out && isFull==false)
       ; // do nothing
    // remove an item from the buffer
    item = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    if(isFull) isFull=false;
    return item;

}
```

3) Do not introduce any new variable (By 张弛):

```
struct item {};
#define BUFFER_SIZE 10
#define BUFFER_SIZE_MASKED (BUFFER_SIZE * 233)
/* As long as the multiplier is larger than 2. Here I choose 233. */

struct item buffer[BUFFER_SIZE];
int in = 0, out = 0;

struct item consumer() {
  while (in == out);
  out = (out + 1) % BUFFER_SIZE_MASKED;
  return buffer[(out - 1) % BUFFER_SIZE];
}

void producer(struct item i) {
  // for example, in = 1, out = 1 + 10 = 11, then the queue is full
  while (in == (out + BUFFER_SIZE) % BUFFER_SIZE_MASKED);
  buffer[in % BUFFER_SIZE] = i;
  in = (in + 1) % BUFFER_SIZE_MASKED;
}
```