```cpp
#include <condition_variable>
#include <iostream>
#include <mutex>
#include <string>
#include <thread>

std::mutex m;
std::condition_variable cv_full, cv_empty;
int data;
bool data_ready;

void consumer_thread(int items) {
  for (int i = 0; i < items; ++i) {
    std::unique_lock<std::mutex> g(m);
    cv_full.wait(g, []() { return data_ready; });
    std::cout << "consumed " << data << std::endl;
    data_ready = false;
    cv_empty.notify_one();
  }
}
void producer_thread(int count, int *items) {
  for (int i = 0; i < count; ++i) {
    std::unique_lock<std::mutex> g(m);
    cv_empty.wait(g, []() { return !data_ready; });
    data = items[i];
    std::cout << "produced " << data << std::endl;
    data_ready = true;
    cv_full.notify_one();
  }
}
int main() {
  int items[] = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55};
  std::thread producer(producer_thread, 10, items);
  std::thread consumer(consumer_thread, 10);
  producer.join();
  consumer.join();
}
```

**FIGURE A.8** Example of using condition variables in C++.

Copyright © 2021 Elsevier Inc. All rights reserved.