Mutex Class Solutions

std::mutex Class

- Briefly describe the C++ std::mutex
  - std::mutex is a class which implements a mutual exclusion object
  - It is defined in <mutex>
  - It has three main member functions
  - lock() tries to lock the mutex. If it is already locked, it waits until the mutex is unlocked, then locks it
  - try_lock tries to lock the mutex. If it is already locked, it returns immediately
  - unlock() unlocks the mutex

Rewrite using std::mutex

- Rewrite the "scrambled output" program using a mutex to protect the output operations
- Verify that the output is not scrambled when there are ten concurrent threads running

try_lock()

- Write a program which runs two task functions in separate threads
- The first task function locks a mutex, sleeps for 500ms and releases the lock
- The second task function sleeps for 100ms, then calls try_lock() to lock the mutex. If unsuccessful, it sleeps again for 100ms and calls try_lock() again. If successful, it unlocks the mutex
- Add suitable print statements and run the program

try_lock()

- What do you observe?
  - Task1 gets the lock first (because of the sleep in Task2)
  - Task2 repeatedly calls try_lock unsuccessfully, because it is locked by Task1
  - Finally, Task1 releases the lock and Task2's try_lock() call succeeds

```
Task1 trying to get lock
Task1 has lock
Task2 trying to get lock
Task2 could not get lock
Task2 could not get lock
Task2 could not get lock
Task2 could not get lock
Task1 releasing lock
Task2 has lock
```