Ch.7\*: Banker’s Algorithm

Write a multithreaded program that implements the *banker’s algorithm* discussed in Section 7.5.3. This assignment combines three topics: (1) multithreading (2) preventing race conditions (3) deadlock avoidance.

Create n customer threads that request and release resources from the bank. The customers will continually loop, requesting and then releasing random numbers of resources. The banker will grant a request if it satisfies the safety algorithm.

Since multiple threads will concurrently access shared data, access must be controlled through mutex locks to prevent race conditions.

You should invoke your program by passing the number of resources of each type on the command line.

做法:由使用者輸入ABC分別的資源總數，並且設定customer會隨機產生所需資源的max以及初始的資源所需，經過隨機的時間後，會完成所有的process且把資源還給CPU，若customer所需資源大於目前CPU可用的資源數量，則會等，等到CPU有足夠的資源才會提供下去。以上customer為不停地loop，而CPU不會進入deadlock。

截圖畫面:

