**Hw3 report**

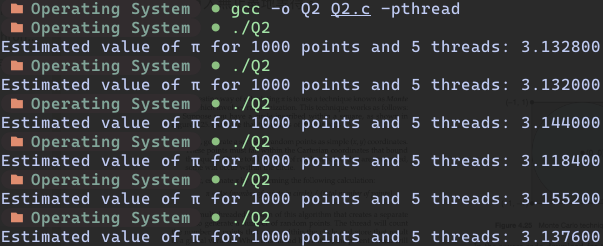
財金所碩二r10723057黃元裕

Q1. 依照 6.2 的題目敘述，證明其演算法滿足三個 critical-section problem 需求

1. Mutual Exclusion: 用flag[ ]和turn來確保mutual exclusion，當2個process的flag皆為TRUE，也只有一個可以進去critical section執行（看turn在誰手上），這邊假設是process i，當process i執行完critical section結束後，會將turn傳給process j，讓j可以進入critical section做事。
2. Progress（一定要有人進去）: 一樣透過flag[ ]和turn來達成，若一個process想進critical section，只要將它的flag[ ]設為TRUE，即可進入critical section做事，且做完結束時會將turn丟給另一個process，換另一個process進去做事。
3. Bounded Waiting（不會有人餓死）: 透過turn來確保，當其中一個的critical section結束後，會將turn傳給另一位，確保不會有人無止盡地等待。

Q2. 題目要求修改原本的單執行緒的程式，設計一個多執行緒的程式，使用 *Mutex Locks* 方法估算圓周率π的值。 原本的程式只有一個執行緒負責生成隨機點，並將結果存儲在一個全域變數 中。在該執行緒結束後，父執行緒進行計算，估算出圓周率的值。 現在需要修改程式，創建多個執行緒，每個執行緒負責生成隨機點並判斷這 些點是否落在圓內。

\*\* 請使用 *Pthreads Mutex Locks*，除了 main thread 外，請再生出 5 個 threads，每個 thread 產生 1000 個 random points 後結束，main thread 最後再 驗收成果且計算出π值。



程式說明：

先建立一個point generated的function，來產生隨機點與判斷點是否在圓裡面，其中建立一個global variable，來進行點的累加。在主程式中，首先設定一個隨機種子seed，使得每次產生的隨機亂數不一樣，再initialize mutex lock，並且生出5個pthread，每個thread中產生1000個random points，並且在Point generated的function中做mutex lock來確保一次只有一個thread在改變global variable，5個thread都完成後回到main計算pi。