

Q1

1. Mutual Exclusion

$flag[i]$ 設成 want-in 表示想進入 critical-section, 之後用 $turn$

協調哪個 process 可以進入, 把 P_i 的 $flag[i] = want-in$, 如果 $flag[turn] \neq idle$ 就檢查下一個 process, 找到 $flag[j] = idle$ 把 $flag[i]$ 設為 in-cs.

P_i 進 critical-section 前檢查所有 $flag$ 確保沒有其他 process 在裡面.

如果 $turn == i$ or $flag[turn] == idle$, P_i 進入 critical-section.

2. Progress

P_i 想進 critical-section 時設 $flag[i] = want-in$

如果 $flag[turn] = idle$ 則 P_i 可以試著進入 critical-section.

P_i 離開 critical-section 之後把 $turn$ 設成下一個非 idle 的 process

最外面的 while 確保所有想進入 critical-section 的 process 都會被等到並循環更新 $turn$.

3. Bounded waiting

$turn$ 確保每個 process 都有公平機會進入 critical-section.

$turn$ 會循環更新, 所以每個 process 都會進入 critical-section.

每個 process 最多等到 $n-1$ 次之後就會進入 critical-section.

Q2

直接執行程式就好, 程式輸出的結果會是計算出來的 pi 值。

```
(base) jiangyanhong@jianm B11705044_Assignment_3 % ./Q2
Estimated value of pi: 3.135200
```