

1.

執行：

```
gcc -pthread -o 01.out first.c
```

```
01.out 90 81 78 95 79 72 85
```

執行程式時，直接傳入要計算的數列

邏輯：

設 Global variables：

cnt：輸入的數字數

avg：平均初始為 0

min：最小值，初始為 INT\_MAX

max：最大值，初始為 INT\_MIN

和 thread：

avg\_func：將傳入的 void\* copy 強制轉換成 int\* copy，計算平均並無條件捨去到整數位

min\_func：將傳入的 void\* copy 強制轉換成 int\* copy，依序判斷數列，比 min 小就更新 min

max\_func：將傳入的 void\* copy 強制轉換成 int\* copy，依序判斷數列，比 max 大就更新 max

Main function 裡，用 argc 數量判斷是否有傳入參數，動態宣告 copy int array 複製跑執行檔時傳入的參數，成數字數列，分別 create 三種 thread，pthread\_join 等待他們結束後輸出。

2.

執行：

```
gcc -pthread -o 02.out second.c
```

```
02.out
```

開始執行程式後才會要求輸入 input (想要產生的點數量)

邏輯：

srand(time(NULL)) 改變系統一開始的亂數值，讓之後出來的亂數和上次執行的不同，設 thread 為 thread identifier，請使用者輸入一個為正整數的值(input)作為即將亂數產生點的數量，pthread\_create function 創 thread 執行 count 的動作，在 count 裡亂數產生 x 與 y 座標界於 0~1 的點共 input 個，這裡使用 (double) rand() / RAND\_MAX，強制轉換作 double 的運算，並計算在圓內的有幾個(到圓心的距離，x 平方與 y 平方相加小於 0)作為 incircle，pthread\_join function 等前述的 thread 做完離開，pi = 4.0 \* incircle / input 輸出。