**Computer Organization 2019**

**HOMEWORK 6**

系級: 資訊111 學號: F74071027 姓名: 學貝萁

**問題(Question)**

Q1. How do you know the number of block from input file?

Q2. How do you know how many set in this cache?

Direct-mapped:

n-way associative:

Fully associative:

Q3. How do you know the bits of the width of the Tag ?

Q4. Briefly describe your data structure of your cache.

使用array來存取資料

Directed-mapped: int cache[set]

|  |  |
| --- | --- |
| index | Tag |
| 0 | -1 (no data) |
| 1 | -1 |

…

4-way associative: int cache[set][4]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| index | Tag1 | Tag2 | Tag3 | Tag4 |
| 0 | -1 | -1 | -1 | -1 |
| 1 | -1 | -1 | -1 | -1 |
| 2 | -1 | -1 | -1 | -1 |

…

Fully associative: int cache[way]

…

|  |  |  |  |
| --- | --- | --- | --- |
| Tag1 | Tag2 | Tag3 | Tag4 |
| -1 | -1 | -1 | -1 |

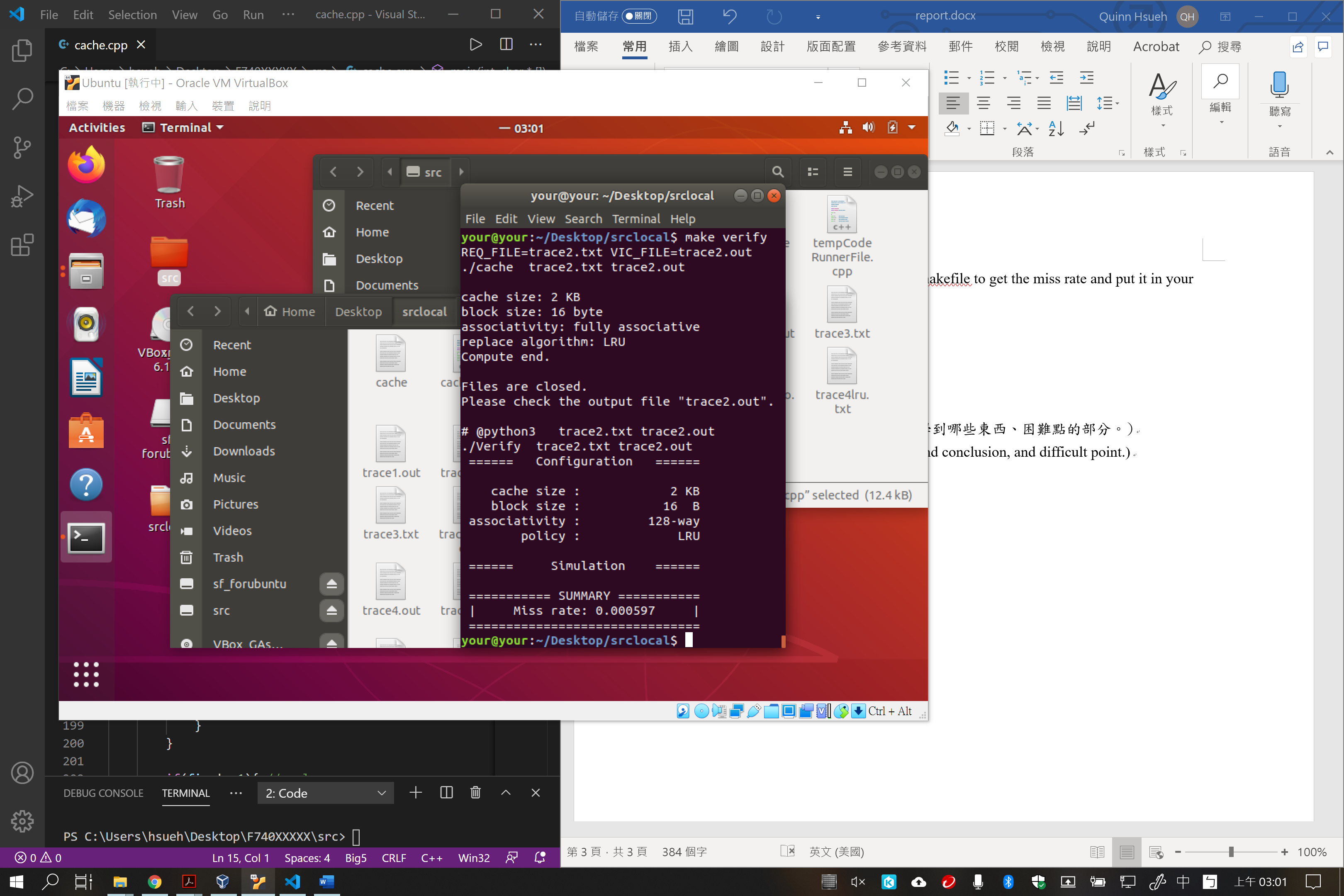
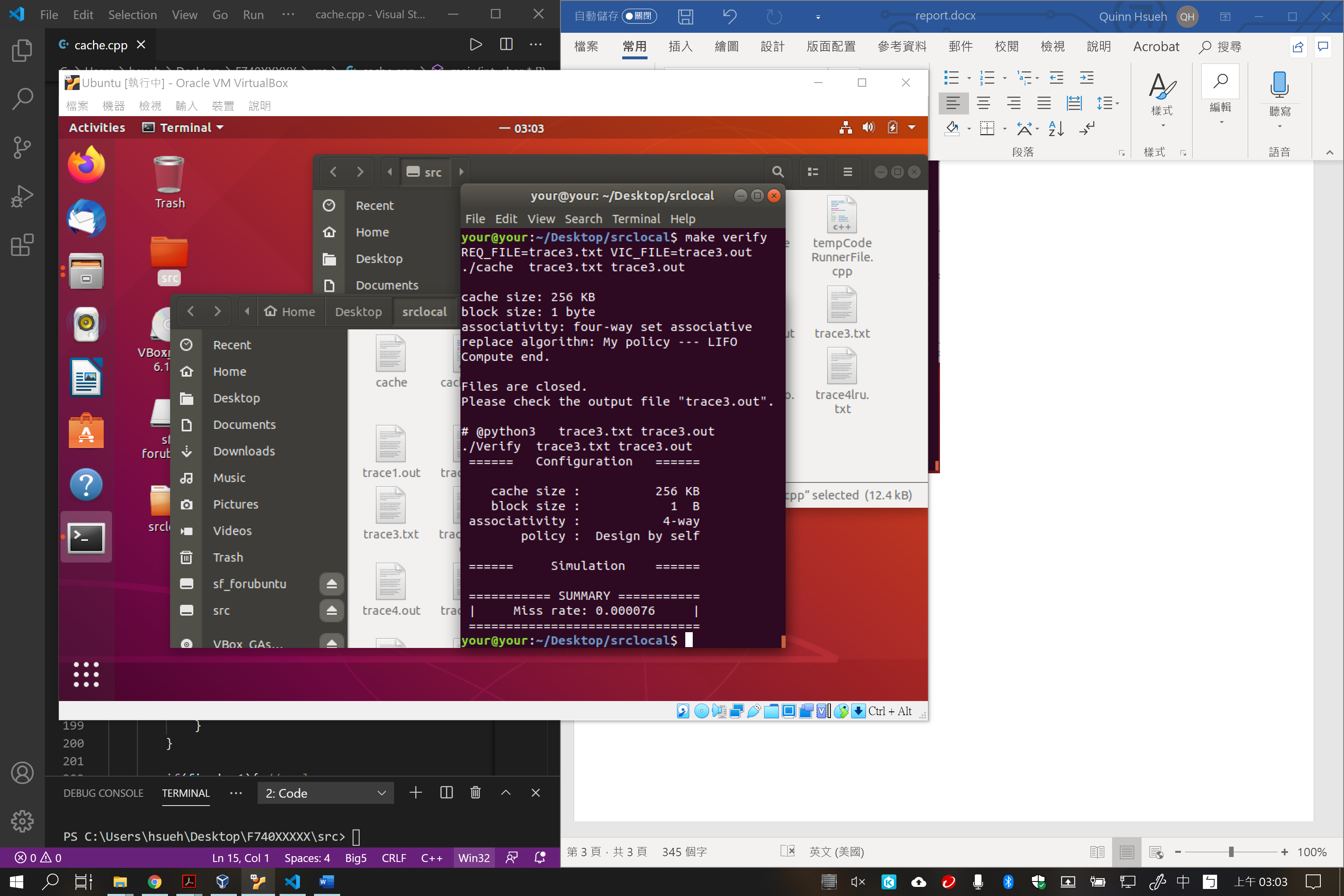
Q5. Briefly describe your algorithm of LRU.

使用list，list的頭是lru，尾是mru。當hit發生時，將發生hit的tag從list移到尾巴(remove再push back)，miss時若set裡還沒滿則直接push back，滿了則將第一個移除(pop front)，並將新的tag push back與在cache array中取代被移除的tag的空間，如此一來，list的順序可以永遠保持lru(降序)。

Q6. Briefly describe your algorithm of your policy.

LIFO(Last In First Out)，使用Stack來完成。

當有資料寫入空白區塊則push，有資料要被取代時，取stack最上層的資料(pop)，並將新資料push進stack。

Q7. Run trace2.txt, trace3.txt and then makefile to get the miss rate and put it in your report.

**心得(Report)**

**透過用程式語言來實作快取記憶體的存取方式，讓我對存取快取資料的流程與順序和種類更有印象。這份程式碼也是近期寫過比較乾淨，而且有認真寫註解的程式碼，所以成就感有大大的增加。**