

HW5

Result screenshot

```
ubuntu@ubuntu2004:~/Desktop/HW5$ gcc -o hw5 hw5.c
ubuntu@ubuntu2004:~/Desktop/HW5$ ./hw5 <input.txt> output.txt
ubuntu@ubuntu2004:~/Desktop/HW5$ cat output.txt
0->77
1->98
2->2
3->37
4->56
5->31
6->45
7->85
8->8
9->70
10->10
11->24
12->64ubuntu@ubuntu2004:~/Desktop/HW5$ ls
ex.txt hw5 hw5.c input.txt output.txt
ubuntu@ubuntu2004:~/Desktop/HW5$ diff ex.txt output.txt
ubuntu@ubuntu2004:~/Desktop/HW5$
```

Program Architecture

先用函式讀入單行的測資，然後用 `atoi` 將其轉成整數儲存，再使用迴圈判斷，根據這次 `hash key` 的原理，將其一一轉入對應的 `hash table` 位置中，底下會詳細解釋。

Program function

```
int atoi(const char *str)
```

這是將讀入的字串轉為整數的函式

How you design your program

這次的程式邏輯還算簡單，可以花不長的時間就想出解法。我的想法是先讀入，並轉為整數儲存後，先在最外面包一層 `for` 由 0 跑到 `TABLE_SIZE`，接著裡面用 `if else` 分成兩個區塊，分別處理目標 `hash table` 是否已經有數字佔據，為了判別，我還額外設了一個 `check` 陣列，長度也是 `TABLE_SIZE`，若該格已經有數字，則該格的 `check` 值為 1，若沒有則為 0。Else if 裡面是處理若目標 `table` 已經有數字，則使用 `do while`，並每次都將 `i+1`，就可以符合 `hash` 規則，這樣這次作業就完成了。