107062208 邱靖豪

Assignment 1

Q1:

0 1 0 2 2

6 0 1 0 0

0 0 0 1 0

7 0 0 0 1

8 0 0 0 1

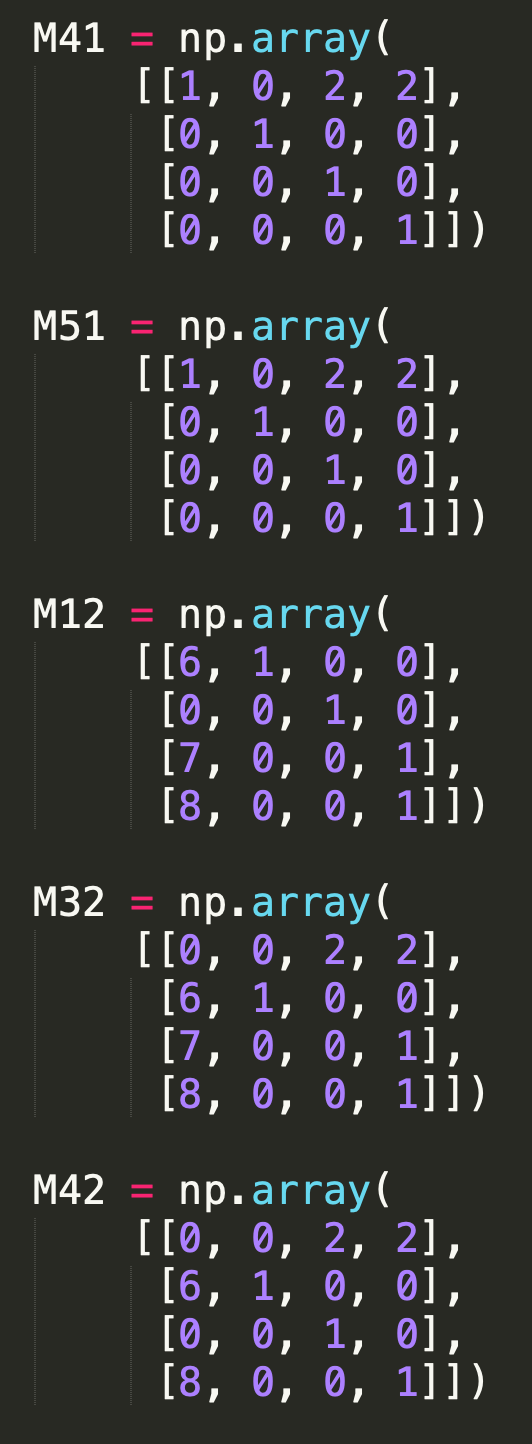
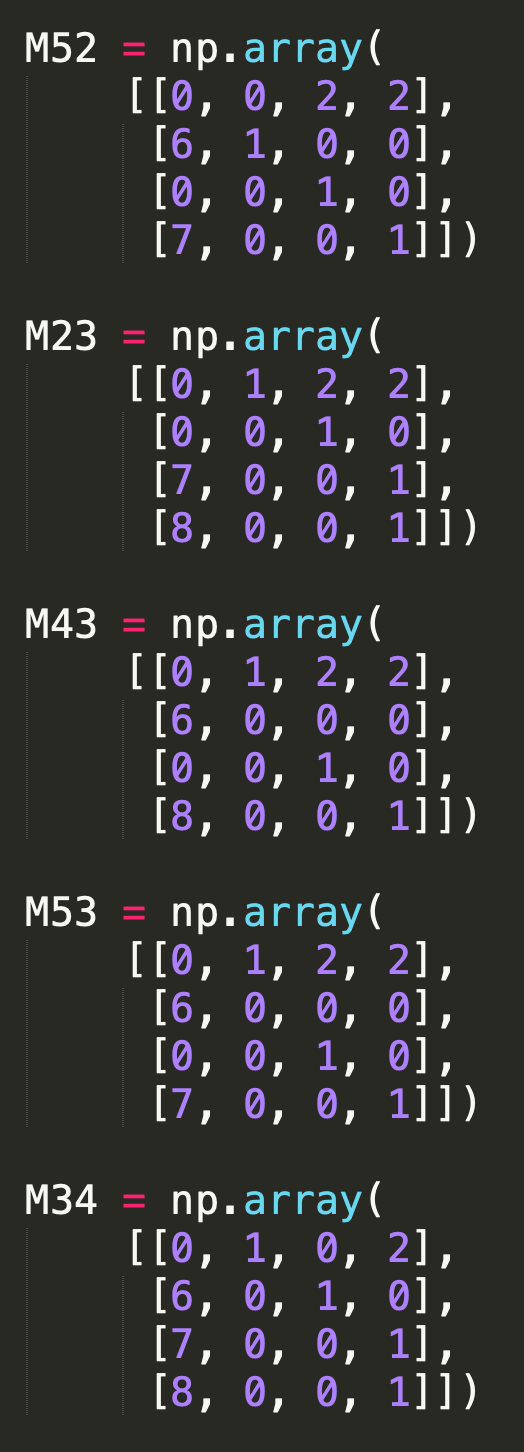
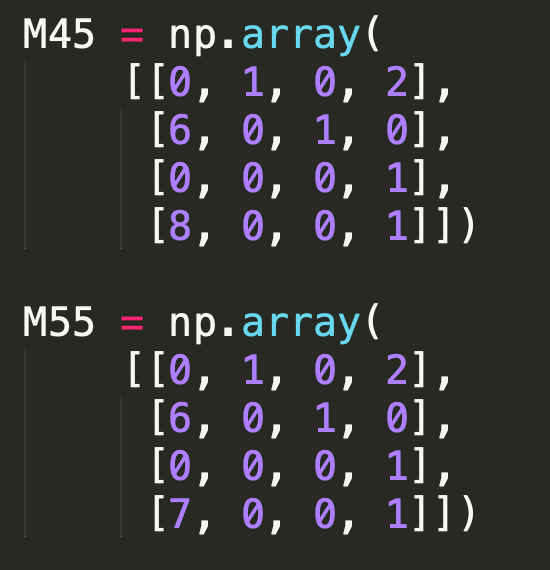
My Hill cipher matrix(A)

如何求A？

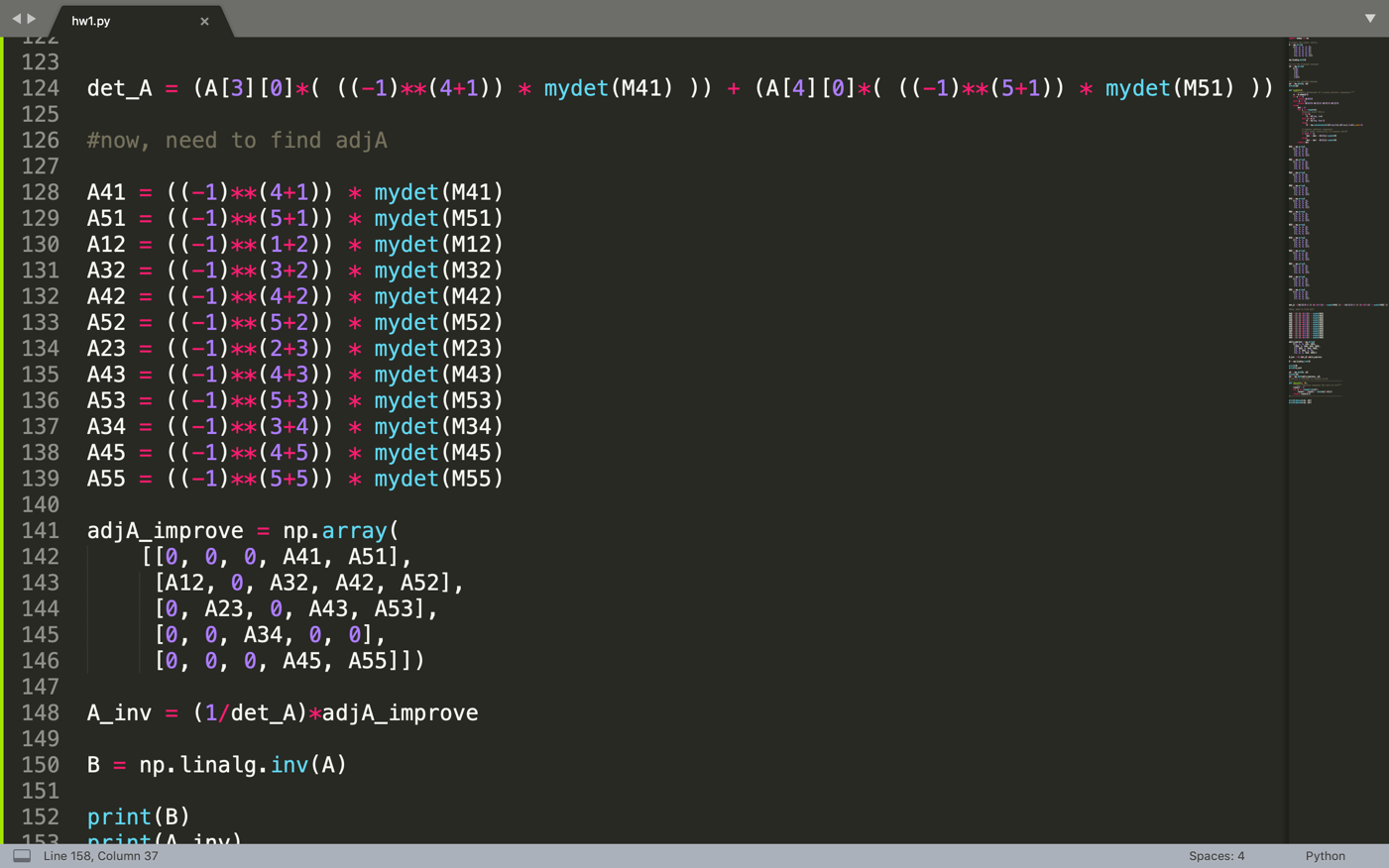
我的學號是107062208，我的想法是利用first row 或first column來做 cofactor expansion，我們可以找出學號中，相差數為1的數，把他們放在first row 或first column，且緊鄰的位置，接著，再讓他們能分出來的M，對角線都是1，接著其他沒用到的位置都補上0，這樣根據我們求det的方法，就能確定得出的行列式值一定是1或-1。以我的學號做說明，我挑出7跟8，並把他們放在a[4][1]跟a[5][1]的位置，所以，我要讓a[1][1]\*A[1][1]，a[2][1]\*A[2][1]， a[3][1]\*A[3][1]都變成 0﹐然後讓 7，8 都乘以一個單位矩陣﹐就可以得 出det(A)為1的矩陣。

Q2:

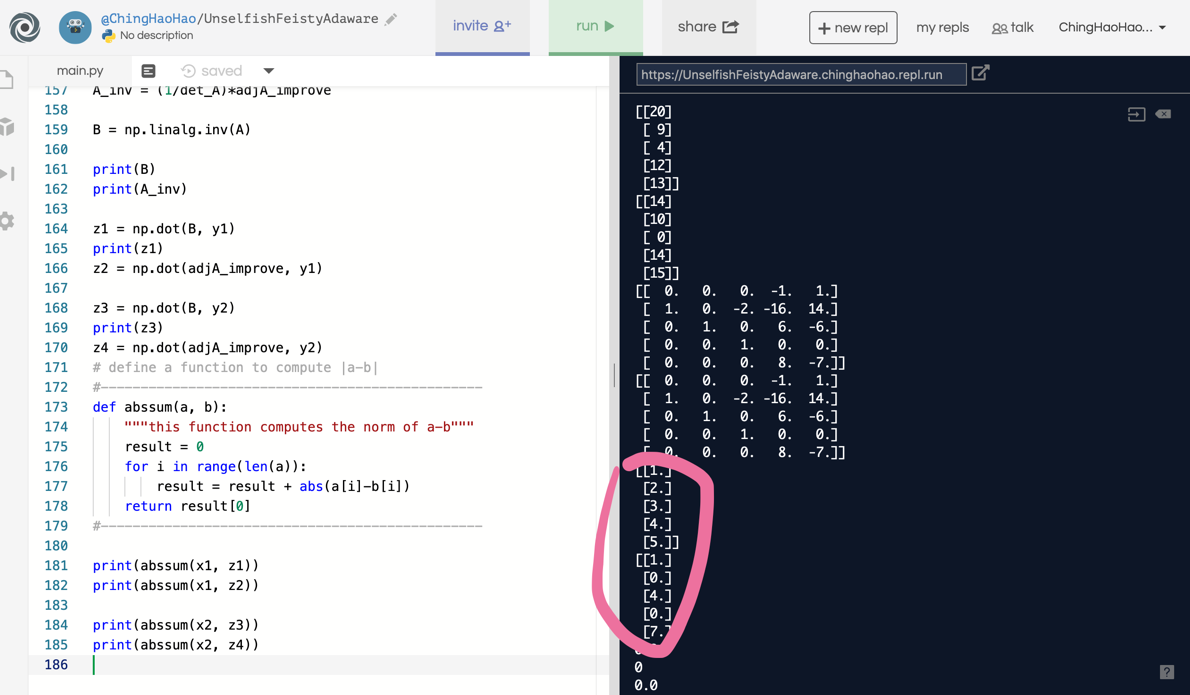
如何求A-1？

我覺得我的想法非常直觀，我是利用講義中給的求adjoint A公式，先找出cofactor Aij，而要找出cofactor Aij，要先求出det(Mij)，Mij的求法我是直接去從我原本的矩陣去消去而來，接著利用教授提供的function，用遞迴找出det。在求M時，我已經先考慮到之後在進行求A-1時，會變成0項的元素了，因此就不列舉出來。接著就可以算det(A)的值。然後，我根據cofactor Aij的公式讓Aij依序被算出，就可以算出adjoint A。



還有經過確認後，正確的加密以及解密結果：



Q3:

Compare original and the decoded message

我們可以發現﹐兩個做法都以 x=[1,2,3,4,5]T 去做 encode 和 decode﹐出來的解果都是對的，在最後 abssum 的地方﹐用 np.linalg.inv(A) 也沒有出現誤差。但是我發現其他同學都有得出誤差值，於是我也去探究原因，原因是因為電腦是用[A|I]這種高斯消去法去做 inverse﹐在做 row operations 時 ﹐ 舉 例 若 同 乘 1/3 ﹐ 電 腦 會 當 作 乘 0.3333333...﹐所以在最後才會產生誤差﹐而 我們用adjoint這題都是整數在做運算﹐因此不會有這些誤差。