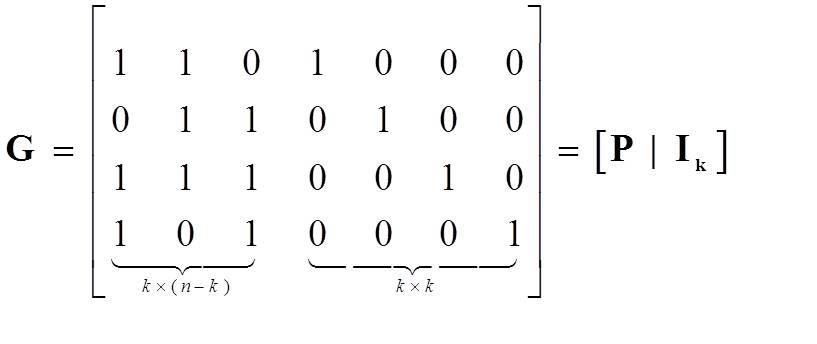
**實習單元二、Linear Block Code**

1. **原理介紹**

一個 (*n,k*) 線性區塊碼，有 *k* 個訊息位元，被編碼成 *n* 個位元，其中有(*n-k*) 個查核位元。(*n,k*) 線性區塊碼的所有字碼形成一個向量空間(亦即在 *Vn(F)* 中由 *2k* 個 *n* 序列所形成之 *k* 維子空間)，它具有一重要特性，即在此空間內，任何兩個向量（在此為碼字）的modulo-2加法運算仍為一個空間內的向量（另一個碼字）。在此令一(*n,k*)線性區塊碼對應於消息序列(information sequence) *u1*= (10...0), *u2* = (01...0), *u3* = (001...0), …, *uk* = (00...01)之字碼分別為 g1 ,g2 , …,gk ，而每一字碼長為 *n*，則其對應之字碼為 ，我們定義生成矩陣(generator matrix)為



從定義可知，一線性段碼 *C* 為 *n* 維向量空間之 *k* 維線性子空間。由線性代數亦可知，如果我們取所有長度為 *n* 且垂直於此 *k* 維線性子空間中所有向量的序列，則可得一個 (*n-k*) 維線性子空間，稱為 *k* 維子空間之正交補集合(orthogonal complement)。這個 (*n-k*) 維子空間本身也定義一個 (*n,n-k*) 線性碼，稱為(*n,k*)原碼 *C* 之重碼(dual code)，寫成 *C⊥*。顯然原碼 *C* 之字碼與重碼 *C⊥* 互相正交。令重碼之生成矩陣為 *H*，則 *H* 為 (*n-k*)*×n* 矩陣，且原碼中任一字碼必正交於*H*中之所有的列，亦即*xHT*=0 for all *x∈C*。而此 *H* 即為重碼 *C⊥* 之生成矩陣，亦為原碼 *C* 之同位檢查矩陣(parity check matrix)，即*GH*T=[*0*]。



1. **程式碼**

這次的作業我使用的軟體是MATLAB，共分為三個部分：

第一部分syndrome table

首先，假設我們已知generator matrix ****，然後由上述內容得知，將generator matrix G 經計算轉換得出 parity check matrix H。我們從得知，只要知道parity check matrix H，就可以知道syndrome。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Error patterns(coset leaders) | | | | | | |
| e0 | e1 | e2 | e3 | e4 | e5 | e6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |

程式碼如下圖2.1所示：

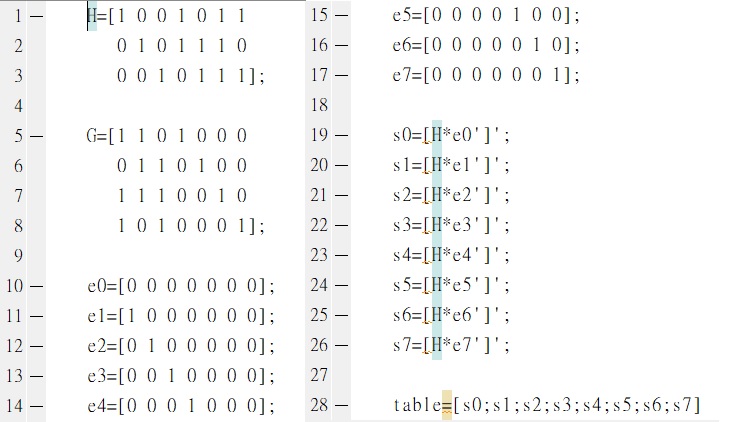


圖2.1

第二部分standard array:

首先，我們先列出2k個字碼於第1列，然後從字碼全為0的C0開始，然後選擇1個最小且在之前陣列中尚未出現的error pattern 排在C0下面，接下來就重複以上步驟直到所有error pattern情況出現。每一列都稱做一個coset，第一行出現的error pattern 稱為coset leader，最後將第1列的字碼分別與各error pattern相加，即可建構出standard array。程式碼如下圖2.2所示：

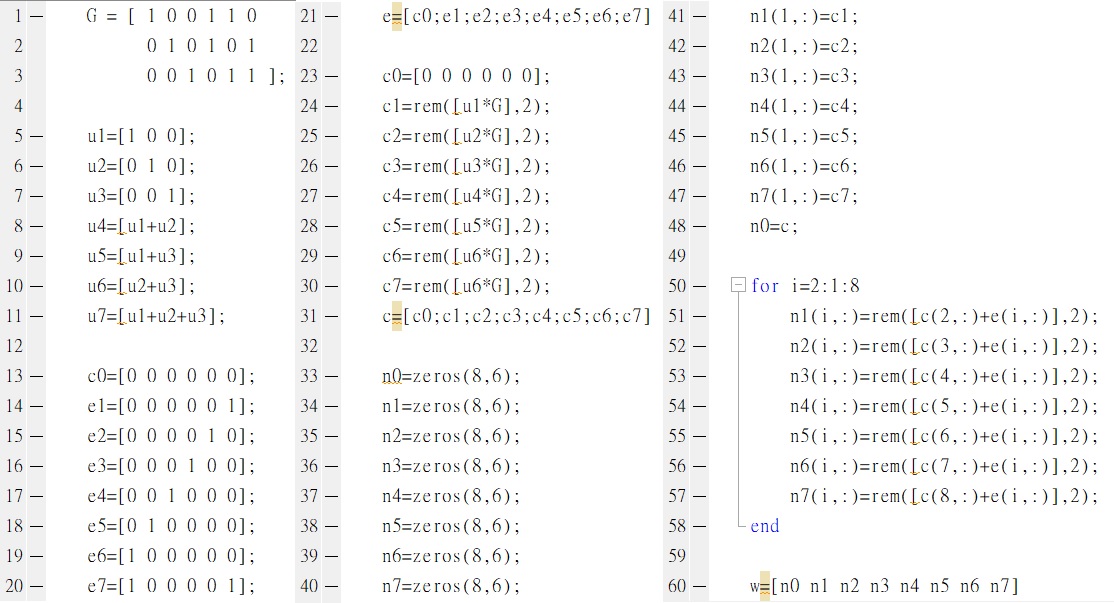


圖2.2

第三部分syndrome decoding

程式碼如下圖2.3所示：

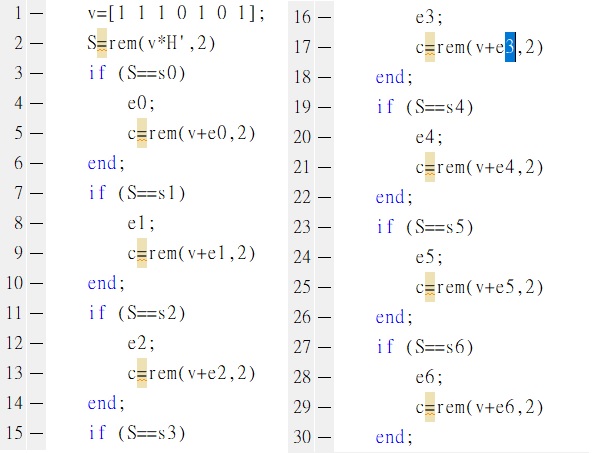


圖2.3

1. **執行結果**

第一部分syndrome table

如圖3.1所示：

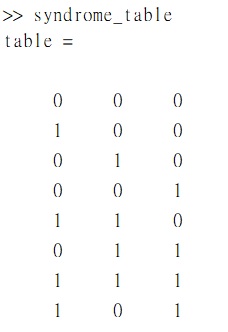


圖3.1

第二部分standard array:

如圖3.2和圖3.3所示：

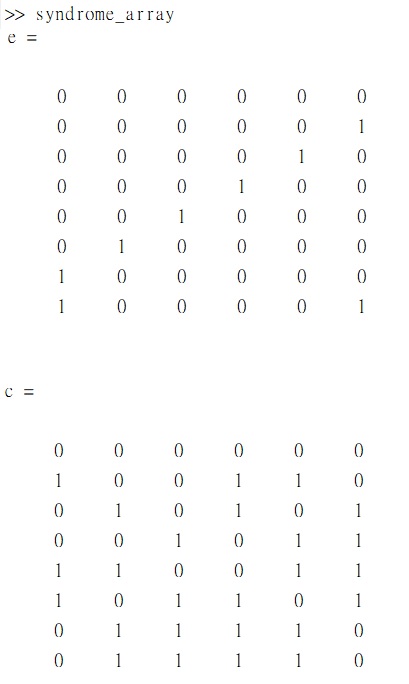


圖3.2

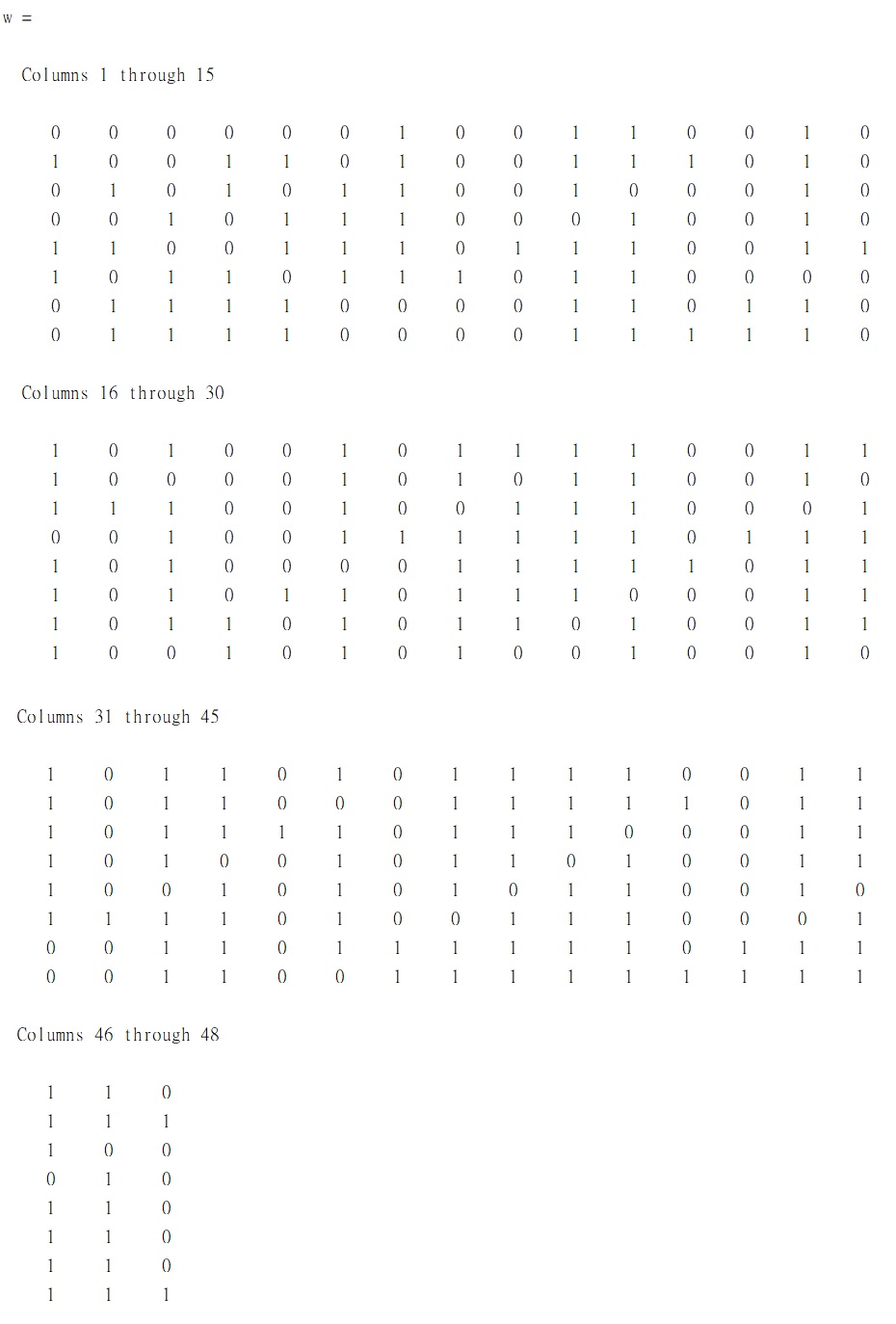


圖3.3

第三部分syndrome decoding

如圖3.4所示：

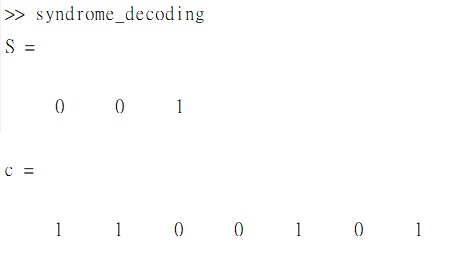


圖3.4