CHAPTER

2

串列

2.1 串列表示法

在未談及串列(array) 之前，讓我們先來看看線性串列(linear list)。線性串列又稱循序串列(sequential list) 或有序串列(ordered list)。其特性是每一項資料是依據它在串列的位置，所形成的一個線性排列，所以 x[i] 會出現在 x[i+1] 之前。

線性串列經常的操作如下：

取出串列中的第 i 項的資料；0 ⭘ i ⭘ n–1。

計算串列的長度。

由左至右或由右至左取出此串列的資料。

在第 i 項加入一個新值，使其原來的第 i，i+1，…，n 項變為第 i+1，i+2，…，n+1 項。亦即在 i 之後的資料都要退後一個位址。

刪除第 i 項，使其原來的第 i+1，i+2，……，n 項變為第 i，i+1，……，n–1 項。亦即在 i 之後的資料都會往前一個位址。

在 Python 程式語言中，我們利用串列來表示線性串列，以線性的對應方式將元素 a_i 置於串列的第 i 個位置上，若要讀取 a_i 時，可利用 $a_i = a_0 + i*d$ 求得。其中 a_0 為串列的起始位址，d 為每一個元素所佔的空間大小。要注意的是，Python 串列的索引或稱註標是從 0 開始。

以下將介紹串列的表示法。

2.1.1　一維串列

若一維串列(one dimension array)是 A(0：u - 1)，而且每一個元素佔 d 個空間，則 **$A(i) = a_0 + i*d$**，其中 a_0 是串列的起始位置。若每一元素，所佔的空間為 d，且起始的元素為 0，則串列 A 的每一元素所對應的位址表示如下：(假設 d=1)

串列元素：	A(0)	A(1)	A(2)	…	A(i)	…	A(u–1)
位　　址：	a_0	a_0+1	a_0+2	…	a_0+(i)	…	a_0+(u–1)

若串列是 A(t：u)，則 **$A(i) = a_0 + (i–t)*d$**。

如串列 A(2：12)，則 A(6) 與 A(2) 起始點相差 4 個單位(6–2)，相當於上述(i–t)。

若串列為 A(1：u)，表示串列的起始元素位置從 1 開始，則 $A(i) = a_0 + (i–1)*d$，其中 d 表示每一元素所佔的空間大小。

範例》

1. 有一串列 A(0：100)，而且 A(0) 的位址是 100，d 為 2，試問 A(16)是多少？

   解 由於 A(i) = 100 + (i)*d

   A(16) = 100 + 16*2 = 132

2. 有一串列 A(–3：10)，而且 A(–3)=100，d=1，求 A(5)=？

   解 由於陣列的形式是(t：u)，所以

   $A(i) = a_0 + (i–t)*d$

   A(5) = 100 + (5–(–3))*1

   = 108

2.1.2　二維串列

假若有一個二維串列(two dimension array) 是 $A[0：u_1–1, 0：u_2–1]$，表示此串列有 u_1 列及 u_2 行；也就是每一列是由 u_2 個元素所組成。二維串列化成一維串列時，對應的方式有二種：(1)以列為主 (row major)，(2)以行為主 (column major)。

1. **以列為主：**視此串列有 u_1 個元素 0，1，2，…，$u_1–1$ ，每一元素有 u_2 個單位，每個單位佔 d 個空間。其情形如圖 2-1 所示

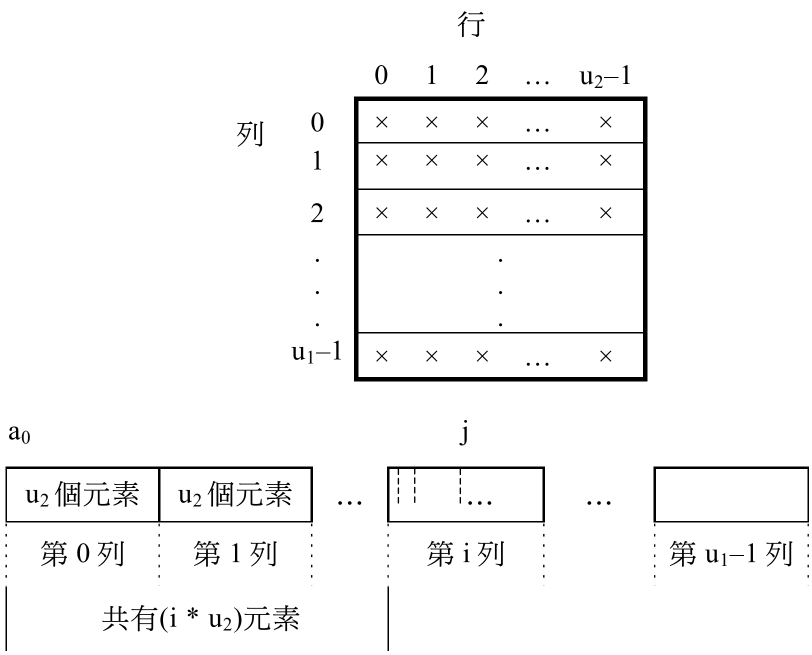


圖 2-1 以列爲主的二維串列循序表示

由上圖可知 **A(i, j) = a_0 + i*u_2*d + j*d**

2. **以行為主**：視此串列有 u_2 個元素 1，2，⋯，u_2，每一元素有 u_1 個單位，每個單位佔 d 個空間。其情形如圖 2-2 所示：

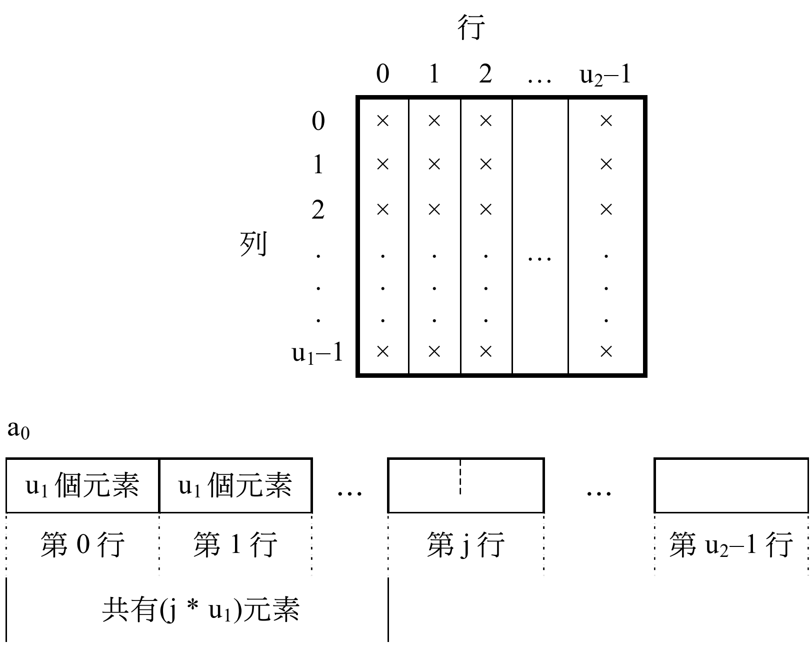


圖 2-2 以行爲主的二維串列循序表示

由上圖可知 $\mathbf{A(i, j) = a_0 + j*u_1*d + i*d}$

假若串列是 $A(s_1：u_1, s_2：u_2)$，則此串列共有 $m = u_1 - s_1 + 1$ 列，$n = u_2 - s_2 + 1$ 行。計算 A(i, j)的位址如下：

1. 以列為主

   $A(i, j) = a_0 + (i-s_1)*n*d + (j-s_2)d$

2. 以行為主

   $A(i, j) = a_0 + (j-s_2)*m*d + (i-s_1)d$

範例》

假設二維串列為 A(–3：5, –4：2)，起始位址是 A(–3, –4) = 100，而且是以列為主排列，請問 A(1, 1) 所在的位址？(d=1)

解 $m = 5 - (-3) + 1 = 9$, $n = 2- (-4) + 1 = 7$, $s_1=-3$, $s_2=-4$, $i=1$, $j=1$

$A(i, j) = a_0 + (i-s_1)*n*d + (j-s_2)d$

$A(1, 1) = 100 + (1-(-3))*7*1 + (1- (-4))1$

$= 100 + 4*7 + 5$

$= 133$

另一解法是將其化為標準式

$A(-3：5, -4：2) \rightarrow A(0：8, 0：6)$，得知 $u_1=9$，$u_2=7$

$A(-3, -4) \rightarrow A(0, 0)$，得知 $A(1, 1) \rightarrow A(4, 5)$

$\therefore A(4, 5) = 100 + 4\times7 + 5 = 100 + 28 + 5 = 133$

2.1.3 三維串列

假若有一個三維串列(three dimension array) 是 $A(0：u_1-1, 0：u_2-1, 0：u_3-1)$，如圖 2-3 所示：

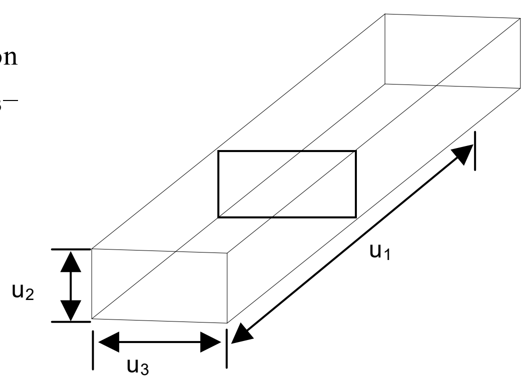
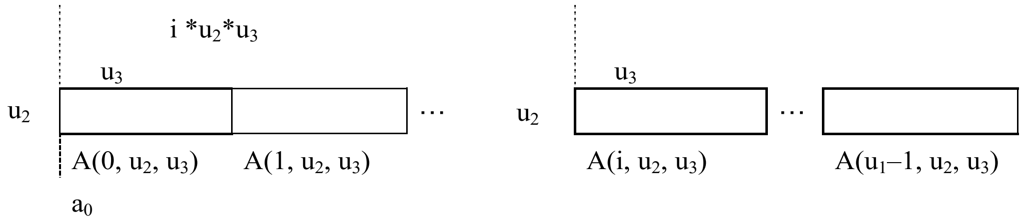


圖 2-3 三維串列以 u_1 個二維串列來表示

一般三維串列皆先轉為二維串列後，再對應到一維串列，對應方式也有二種：(1)以列為主，(2)以行為主。

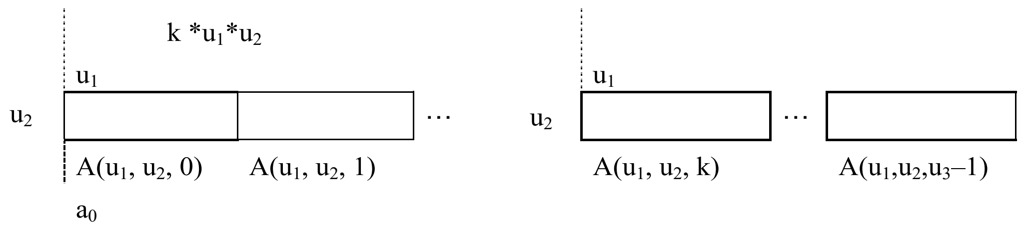
1. **以列為主**

   視此串列有 u_1 個 u_2*u_3 的二維串列，每一個二維串列有 u_2 個元素，每個 u_2 皆有 u_3d 個空間。



   $\mathbf{A(i\ ,\ j\ ,\ k) = a_0 + i*u_2*u_3*d + j*u_3*d + k*d}$

2. **以行為主**



   $\mathbf{A(i\ ,\ j\ ,\ k) = a_0 + k*u_1*u_2*d + j*u_1*d + i*d}$

   假設串列為 $A(s_1：u_1, s_2：u_2, s_3：u_3)$，則 $p = u_1 - s_1+1$，$q = u_2 - s_2+1$，$r = u_3 - s_3+1$。

   以列為主的公式為：

   $A(i\ ,\ j\ ,\ k) = a_0 + (i-s_1)*q*r*d + (j-s_2)*r*d + (k-s_3)*d$

   以行為主的公式為：

   $A(i\ ,\ j\ ,\ k) = a_0 + (k-s_3)*p*q*d + (j-s_2)*p*d + (i-s_1)*d$

範例》

假設有一三維串列 A(–3：5, –4：2, 1：5)，其起始位址為 A(–3, –4, 1) = 100，且是以列為主的排列，試求 A(1, 1, 3)所在的位址？(d=1)

解 $p = 5-(-3) + 1 = 9,\ q = 2-(-4) + 1 = 7,\ r = 5 - 1 + 1 = 5,$

$s_1=-3,\ s_2=-4,\ s_3=1;\ i=1,\ j=1,\ k=3$

$A(1, 1, 3) = 100 + (1-(-3)) * 7*5*1 + (1-(-4)) *5*1 + (3-1)*1 = 267$

另一種計算方式是為將式子化為標準式來看

$A(-3:5, -4:2, 1:5) \rightarrow A(0:8, 0:6, 0:4)$__________

從上式得知 $u_1=9$，$u_2=7$，$u_3=5$

$A(1, 1, 3) \rightarrow A(4, 5, 2)$__________

此乃第　式化為標準式，將(-3, -4, 1) 分別加 3，加 4 及減 1 的原故，所以也要對(1, 1, 3) 加 3，加 4 及減 1 的動作，使其成為第　式。同時 $A(-3, -4, 1) \rightarrow A(0, 0, 0)=100$ (已知)

$$\begin{aligned}\therefore A(4, 5, 2) &= 100 + 4*7*5 + 5*5 + 2 \\ &= 100 + 140 + 25 + 2 \\ &= 267\end{aligned}$$

此答案與上一種解法所求出的答案是相同。

2.1.4　n 維串列

若有一 n 維串列(n dimension array)為 $A(0:u_1-1, 0:u_2-1, 0:u_3-1, \cdots, 0:u_n-1)$，表示 A 串列為 n 維串列，同樣 n 維串列亦有二種表示方式：(1) 以列為主，(2) 以行為主。

1. **以列為主：**若 A 串列以列為主，表示 A 串列有 u_1 個 n–1 維串列，u_2 個 n–2 維串列，u_3 個 n–3 維串列，…及 u_n 個一維串列。假設起始位址為 a_0，則

$A(0, 0, 0, \cdots, 0)$之位址為　a_0

$A(i_1, 0, 0, \cdots, 0)$之位址為　$a_0 + i_1 * u_2 u_3 \cdots u_n$

$A(i_1, i_2, 0, \cdots, 0)$ 之位址為　$a_0 + i_1 * u_2 u_3 \cdots u_n + i_2 * u_3 u_4 \cdots u_n$

…

$A(i_1, i_2, i_3, \cdots, i_n)$之位址為

$$\begin{aligned} & a_0 + i_1 * u_2 u_3 \cdots u_n \\ & + i_2 * u_3 u_4 \cdots u_n \\ & + i_3 * u_4 u_5 \cdots u_n \\ & \cdots \\ & + i_{n-1} * u_n \\ & + i_n \end{aligned}$$

上述可歸納為：

$\mathbf{A(i_1, i_2, i_3, \cdots, i_n) = a_0 + \sum_{m=1}^{n} i_m * a_m}$，其中

$$\begin{cases} a_m = \prod_{p=m+1}^{n} u_p, 1 \le m < n \\ a_n = 1 \end{cases}$$

(此處的 π 是連乘的意思)

2. **以行為主：**若 A 串列以行為主，表示 A 串列有 u_n 個 n–1 維串列，u_{n-1} 個 n–2 維串列，⋯，u_j 個 j–1 維串列及 u_2 個一維串列。假設起始位址亦是 a_0，則

$A(0, 0, 0, \cdots, 0)$ 之位址為 a_0

$A(0, 0, 0, \cdots, i_n)$ 之位址為 $a_0 + i_n * u_1 u_2 \cdots u_{n-1}$

$A(0, 0, 0, \cdots, i_{n-1}, i_n)$ 之位址為 $a_0 + i_n * u_1 u_2 \cdots u_{n-1} + i_{n-1} * u_1 u_2 \cdots u_{n-2}$

⋯

$A(i_1, i_2, i_3, \cdots, i_n)$ 之位址為 $a_0 + i_n * u_1 u_2 \cdots u_{n-1}$
$+ i_{n-1} * u_1 u_2 \cdots u_{n-2}$
$+ i_{n-2} * u_1 u_2 \cdots u_{n-3}$
⋯
$+ i_2 * u_1$
$+ i_1$

上述可歸納為：

$A(i_1, i_2, i_3, \cdots, i_n) = a_0 + \sum_{m=1}^{n} i_m * a_m$，其中

$$\begin{cases} a_m = \prod_{p=1}^{m-1} u_p, 2 \le m < n \\ a_1 = 1 \end{cases}$$

練習題

1. 有一個二維串列如下：$A(1：u_1, 1：u_2)$，若分別寫出(a)以列為主(b)以行為主的 A(i, j)=？

2. 假設 A(–3：5, –4：2)且其起始位址 A(–3, –4) = 100，以行為主排列，請問 A(1, 1)所在的位址？(d=1)

3. 若有一串列為 A(1：u_1, 1：u_2, 1：u_3)，試問 A(i, j, k)分別以列為主，和以行為主所在的位址各為何？

4. 假設有一個三維串列 A(–3：5, –4：2, 1：5)，其起始位址為 A(–3, –4, 1) =100，而且是以行為主排列，則 A(2, 1, 2)所在的位址為何？(d=1)

5. 假設有一個 n 維串列 A(1：u_1, 1：u_2, 1：u_3..., 1：u_n)，試分別寫出以列為主和以行為主的 A(i_1, i_2, i_3, ..., i_n) 之位址為何？

2.2 Python 語言的串列表示法

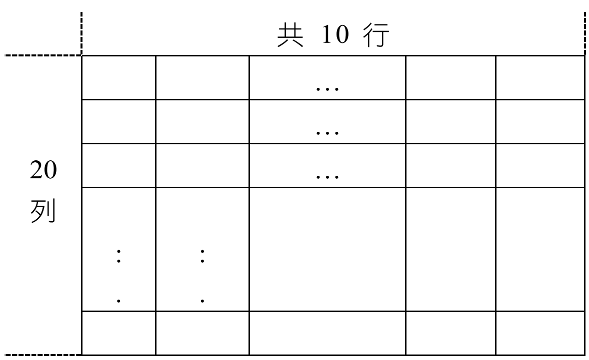
Python 語言的一維串列表示如下：

```
A = [0] * 20
```

表示 A 串列有 20 個整數元素，從 A[0]到 A[19]。注意！Python 語言的串列註標起始值為 0，而二維串列表示方法為

```
A = [[0] * 10 for rows in range(20)]
```

表示 A 串列有 20 列、10 行，如下圖所示：

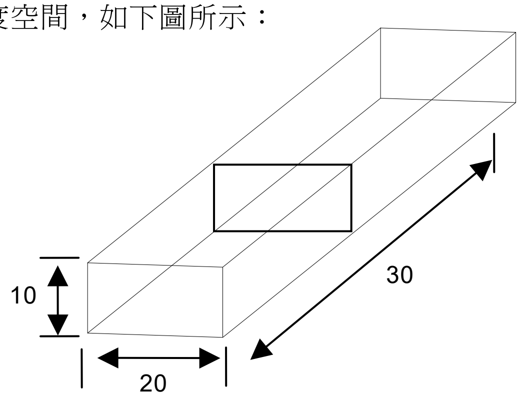


從 A[0][0]，A[0][1]，⋯，A[19][9]等 200 個元素，其初始值皆為 0。

以此類推，三維串列就是由三個中括號表示之，如

```
A = [[[0 for z in range(10)] for x in range(20)] for y in range(30)]
```

其所表示的圖形就像三度空間，如下圖所示：



此串列共有 6000 個元素，每一元素的初始值為 0。

練習題

1. 試問下列片段程式的輸出結果為何？

(a)

```
i = []
total=0
for k in range(10):
    i.append(k+1)
for k in range(10):
    total += i[k]
print("%d" % total)
```

(b)

```
arr2 = []
total = 0
for i in range(5):
    new = []
    for j in range(5):
        new.append(i + j)
        print("%3d" % new[j], end = '')
        arr2.append(new)
   print()

for i in range(5):
    for j in range(5):
        total += arr2[i][j]
print("\ntotal = %d" % total)
```

2. 將第一章動動腦時間的第 2 題，實際以 Python 程式來執行，並檢查你做的答案是否與它相符合。

2.3 矩陣

1. 矩陣相乘

假設 A = (a_{ij})是一 m*n 的矩陣，而 B = (b_{ij})為 n*s 的矩陣，則 AB 的乘積為 m*s 的矩陣

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

如下圖所示：

$$\underset{m \times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & & \vdots \\ \hline a_{i1} & a_{i2} & \cdots & a_{in} \\ \hline \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}} \underset{n \times s}{\left[\begin{array}{c|c|c} b_{11} \cdots & b_{1j} & \cdots \; b_{1s} \\ b_{21} \cdots & b_{2j} & \cdots \; b_{2s} \\ \vdots & \vdots & \vdots \\ b_{n1} \cdots & b_{nj} & \cdots \; b_{ns} \end{array}\right]} \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1s} \\ \vdots & & & & \\ c_{i1} & \cdots & \boxed{c_{ij}} & \cdots & c_{is} \\ \vdots & & & & \\ c_{m1} & \cdots & c_{mj} & \cdots & c_{ms} \end{bmatrix}$$

表示 c 串列的第 i 列第 j 行的值為 a 串列的第 i 列乘以 b 串列的第 j 行

即 $C_{ij} = a_{i1}*b_{1j}+a_{i2}*b_{2j}+\cdots+a_{in}*b_{nj}$

範例》

$$\begin{bmatrix} 2 & 1 & -3 \\ -2 & 2 & 4 \end{bmatrix} \begin{bmatrix} -1 & 2 \\ 0 & -3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2(-1)+1(0)+(-3)2 & 2(2)+1(-3)+(-3)1 \\ (-2)(-1)+2(0)+4(2) & (-2)2+2(-3)+4(1) \end{bmatrix}$$

$$= \begin{bmatrix} -8 & -2 \\ 10 & -6 \end{bmatrix}$$

矩陣相乘的片段程式如下：

Python 片段程式》 矩陣相乘

```
def matrix(A, B, N):
    # 將 A 矩陣每一列元素與 B 矩陣每一列元素
    # 相乘之和放入 C 矩陣之中
    for i in range(N):
        for j in range(N):
            sum = 0
            for k in range(N):
```

```
        sum += A[i][k] * B[k][j]
    C[i][j] = sum
```

有關矩陣相乘之程式實作，請參閱 2.8 節。

2. **稀疏矩陣**

若一矩陣中有大多數元素為 0 時，則稱此矩陣為稀疏矩陣(sparse matrix)，到底要多少個 0 才算是疏稀，則沒有絕對的定義，一般而言，大於 1/2 個就可稱之，如圖 2-4 是一稀疏矩陣。

$$\begin{bmatrix} 0 & 15 & 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 18 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 \\ 72 & 0 & 0 & 0 & 20 & 0 \end{bmatrix}$$

圖 2-4 稀疏矩陣

此矩陣共有 36 個元素，但只有 7 個元素非為 0，因此 0 的個數佔了 80%左右，若是將 1000×1000 矩陣以二維串列來儲存，勢必會浪費許多空間，因為這些 0 根本不必管它，只要儲存非零的元素即可，因此，我們可以下列的資料結構表示之。

(i, j, value)，其中 i 表示第幾列，j 表示第幾行，而 value 表示要儲存的值，若將此矩陣的非零值儲存於二維串列 A(0：n, 1：3) 的結構中，其中 n 表示非零的數字。上一矩陣若以此結構表示的話，情況如下所示(假設二維串列的第一個元素為 A(1, 1))：

	1)	2)	3)
A(0,	6	6	7
A(1,	1	2	15
A(2,	1	5	–8
A(3,	3	4	–6
A(4,	4	3	18
A(5,	5	6	16
A(6,	6	1	72
A(7,	6	5	20

某一細胞若相鄰的活細胞為 2 或 3，則它將會存活下來。但圖中的死細胞其相鄰的活細胞都是小於或等於 2，故不能再生，所以已成為穩定狀態。

範例三》

0	0	0	0	0
1	2	3	2	1
1	@1	@2	@1	1
1	2	3	2	1
0	0	0	0	0

根據上述規則，它的下一代將為

0	1	1	1	0
0	2	@1	2	0
0	3	@2	3	0
0	2	@1	2	0
0	1	1	1	0

這二張圖將會來回的互換。

有關生命細胞遊戲的程式實作，請參閱 2.8 節。

2.8 程式實作

(一) 矩陣相乘

Python 程式語言實作》兩個矩陣相乘

```
# 矩陣相乘實作
# 將兩矩陣行列相乘之和放入第三個矩陣
# File Name: matrixMultiply.py
# Version 4.0 (Updated on May 8, 2021)

N = 5
C = [[0] * N for row in range(N)]

def access_matrix(A, B):
```

```
    global C

    # 將A 矩陣每一列元素與B 矩陣每一列元素
    # 相乘之和放入C 矩陣之中
    for i in range(N):
        for j in range(N):
            sum = 0
            for k in range(N):
                sum += A[i][k] * B[k][j]
            C[i][j] = sum

def output_result(A, B):
    global C

    # 列出三矩陣內容
    print("\nContent of Matrix A :\n")
    output_matrix(A)
    print("\nContent of Matrix B :\n")
    output_matrix(B)
    print("\nContent of Matrix C :\n")
    output_matrix(C)

def output_matrix(m): # 輸出陣列內容
    for i in range(N):
        for j in range(N):
            print(" ", m[i][j], end = '')
        print() # 輸出完一列跳行

def main(): # 主函數
    A = [[0]*N for row in range(N)] # 宣告5x5 陣列A,並將所有元素指定為0
    B = [[0]*N for row in range(N)] # 宣告5x5 陣列B,並將所有元素指定為0

    for i in range(N):
        for j in range(N):
            A[i][j] = j + 1 # 給5x5 的陣列A 指定初始值
    for i in range(N):
        for j in range(N):
            B[i][j] = -(j - 5) # 給5x5 的陣列B 指定初始值

    access_matrix(A, B)
    output_result(A, B)

main()
```

輸出結果

```
Content of Matrix A :

  1  2  3  4  5
  1  2  3  4  5
  1  2  3  4  5
  1  2  3  4  5
  1  2  3  4  5

Content of Matrix B :

  5  4  3  2  1
  5  4  3  2  1
  5  4  3  2  1
  5  4  3  2  1
  5  4  3  2  1

Content of Matrix C :

  75  60  45  30  15
  75  60  45  30  15
  75  60  45  30  15
  75  60  45  30  15
  75  60  45  30  15
```

(二) 多項式相加

Python 程式語言實作》多項式相加－利用串列表示法做多項式相加

```
# 多項式相加實作
# 利用陣列表示法做多項式相加
# File Name: polynomialAdd.py
# Version 4.0 (Updated on May 8, 2021)

DUMMY = -1

def padd(a, b, c):
    p = q = r = m = n = 0
    m = a[1]
    n = b[1]
    p = q = r = 2

    while p <= 2*m and q <= 2*n:
        # 比較a 與b 的指數
        result = compare(a[p], b[q])

```

```
------------------------
------------------------
Continue next Generation ? (y/n): y
Game of life cell status
------Generation 5------

------------------------
---------@@@------------
--------@---@-----------
--------@---@-----------
--------@---@-----------
---------@@@------------
------------------------
------------------------
------------------------
------------------------
Continue next Generation ? (y/n): n
```

2.9 動動腦時間

1. 假設有一串列 A，其 A(0, 0)與 A(2, 2) 的位址分別在$(1204)_8$ 與$(1244)_8$，求 A(3, 3) 的位址(以 8 進位表示)。[2.1]

2. 有一三維串列 A(–3：2, –2：4, 0：3)，以列為主排列，串列的起始位址是 318，試求 A(1, 3, 2) 所在的位址。[2.1]

3. 有一二維串列 A(0：m–1, 0：n–1)，假設 A(3, 2) 在 1110，而 A(2, 3)在 1115，若每個元素佔一個空間，請問 A(1, 4) 所在的位址。[2.1]

4. 若將一對稱矩陣(symmetric matrix) 視為上三角形矩陣來儲存，亦即 a_{11} 儲存在 A(1)，$a_{12} = a_{21}$ 儲存在 A(2)，a_{22} 在 A(3)，$a_{13} = a_{31}$ 在 A(4)，$a_{23} = a_{32}$ 在 A(5)，及 a_{ij} 在 A(k)地方。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix}$$

試求 A(i, j) 儲存的位址(可用 MAX 與 MIN 函數來表示，其中 MAX 函數表示取 i, j 的最大值，MIN 函數則是取 i, j 最小值。)[2.5]

5. 有一正方形矩陣，其存放在一維串列的形式如下：

$$\begin{bmatrix} A(1) & A(2) & A(5) & A(10) & \dots \\ A(4) & A(3) & A(6) & A(11) & \dots \\ A(9) & A(8) & A(7) & A(12) & \dots \\ A(16) & A(15) & A(14) & A(13) & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \end{bmatrix}$$

讓 a_{ij} 儲存在 A(k)，試求 A(i, j) 所在的位址，可用 MAX 及 MIN 函數來表示。[2.5]

6. 試回答下列問題：[2.5]

(a) 撰寫一演算法將 A_{nxn} 的下三角形儲存於一個 B(1：n(n+1)//2) 的串列中

(b) 撰寫一演算法從上述的串列 B 中取出 A(i, j)

7. 在 2.6 節我們談到一個很有趣的魔術方陣，首先在第一列的中間填上 1，之後往左上方走，再遵循一些規則便可完成。如今，若改變方向，填上 1 之後，往右上方走，是否也可以完成魔術方陣呢？略述您的規則。[2.6]

8. 試完成下列生命細胞遊戲。[2.7]

(a)

			@			
				@		
		@	@	@		

(b)

			@			
		@	@	@		