

# 實驗9-12

- 9. 十進位轉七進位
- 10. 十進位轉十七進位
- 11. 最大質因數
- 12. 二維陣列生成

**Ex9. (25 points)** 隨機產生小於1023  
的正整數，使用**while looping**  
求該代表該正整數的七進位字串，  
並印出字串

While迴圈的進入條件要如何設計呢？

```
import random
N = 100000
S = list(range(1,N))
n = random.choice(S)
print(bin(n))
print("converting",n," to")
b = ""
while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2
print("0b"+b)
```

While迴圈和for迴圈有什麼不一樣呢？

While迴圈中的主體命令要如何設計呢？

為什麼需要使用while迴圈？難道不能只使用for迴圈，就好了呢？

```
0b10100100001000110
converting 84038 to
0b10100100001000110
```

```

while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2

```

迴圈	進入條件 n>0	r	n	b
初始值			11	""
迴圈1	T	1	5	"1"
迴圈2	T	1	2	"11"
迴圈3	T	0	1	"011"
迴圈4	T	1	0	"1011"
迴圈5	F			

```

b = ""
n = 11
while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2
    print(n>0, " r:",r," n:",n," b:",b)
print("0b"+b)

```

```

True r: 1 n: 5 b: 1
True r: 1 n: 2 b: 11
True r: 0 n: 1 b: 011
False r: 1 n: 0 b: 1011
0b1011

```

迴圈	進入條件 n>0	r	n	b
初始值			11	""
迴圈1	T	1	5	"1"
迴圈2	T	1	2	"11"
迴圈3	T	0	1	"011"
迴圈4	T	1	0	"1011"
迴圈5	F			

```

b = ""
n = 23
while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2
    print(n>0, " r:",r, " n:",n, " b:",b)
print("0b"+b)

```

```

True  r: 1  n: 11  b: 1
True  r: 1  n: 5   b: 11
True  r: 1  n: 2   b: 111
True  r: 0  n: 1   b: 0111
False r: 1  n: 0   b: 10111
0b10111

```

迴圈	進入條件 n>0	r	n	b
初始值			23	""
迴圈1	T			""
迴圈2				""
迴圈3				""
迴圈4				""
迴圈5				

**Ex10A. (25 points)** 隨機產生小於  
**1024**的正整數，使用**while looping**  
求該代表該正整數的十七進位字  
串，並印出字串

# 十進位轉十六進位提示

- 參考十進位轉二進位的程式
- 在while迴圈中有兩個命令使用底數2，本實驗需將底數改為16

```
while n > 0:  
    r = n % 2  
    b = str(r) + b  
    n = (n - r) // 2
```

轉二進位  
使用底數  
為2

# 十進位轉十六進位提示

- 餘數 $r$ 介於1-15
- 如果餘數小10，直接皆將 $r$ 轉為字串，附加在 $b$ 字串的左邊
- 如果餘數大於等於10，則參考下表在 $b$ 字串的左邊附加字串

10	"A"
11	"B"
12	"C"
13	"D"
14	"E"
15	"F"

```
while n > 0:  
    r = n % 2  
    b = str(r) + b  
    n = (n - r) // 2
```

餘數 $r$   
為0與1

**Ex10B. (25 points) 使用while  
looping 將Ex8A產生的  
十七進位字串，轉為十進位，  
並印出**

**EX11A. (10 points)** 從1-99999中  
隨機產生一個正整數，如果不是質  
數，印出一個最大質因數與其他數  
相乘的式子，否則印出'**PRIME**'

## 參考範例 slides 22-28

從1-99999中隨機產生一個正整數，如果不是質數，印出一個最小質因數與其他數相乘的式子，否則印出'**PRIME**'

# 提示

- 參考範例是找正整數 $n$ 的最小質因數，for迴圈中的`range(2,n)`的設計，是從2開始找，每次增加1，遞增檢查所有小於 $n$ 的數
- 本題要找 $n$ 的最大質因數，可以改變`range`的設計，從最大的數，每次遞減1，遞減檢查所有大於1的數

**EX11B. (20 points) 從1-99999  
中隨機產生一個正整數，如果  
不是質數，印出所有質因數，  
否則印出'PRIME'**

# Ex12A. (10 points) 使用range, reshape, transpose 分別產生以下的二維陣列

```
[[ 1  2  3  4  5  6  7  8  9 10]
 [11 12 13 14 15 16 17 18 19 20]
 [21 22 23 24 25 26 27 28 29 30]
 [31 32 33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48 49 50]
 [51 52 53 54 55 56 57 58 59 60]
 [61 62 63 64 65 66 67 68 69 70]
 [71 72 73 74 75 76 77 78 79 80]
 [81 82 83 84 85 86 87 88 89 90]
 [91 92 93 94 95 96 97 98 99 100]]
```

```
[[ 1 11 21 31 41 51 61 71 81 91]
 [ 2 12 22 32 42 52 62 72 82 92]
 [ 3 13 23 33 43 53 63 73 83 93]
 [ 4 14 24 34 44 54 64 74 84 94]
 [ 5 15 25 35 45 55 65 75 85 95]
 [ 6 16 26 36 46 56 66 76 86 96]
 [ 7 17 27 37 47 57 67 77 87 97]
 [ 8 18 28 38 48 58 68 78 88 98]
 [ 9 19 29 39 49 59 69 79 89 99]
 [10 20 30 40 50 60 70 80 90 100]]
```

## 參考範例 slides 6-12

# 使用range, reshape 生成二維陣列

---

```
[[ 1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18]
 [19 20 21 22 23 24 25 26 27]
 [28 29 30 31 32 33 34 35 36]
 [37 38 39 40 41 42 43 44 45]
 [46 47 48 49 50 51 52 53 54]
 [55 56 57 58 59 60 61 62 63]
 [64 65 66 67 68 69 70 71 72]
 [73 74 75 76 77 78 79 80 81]]
```

# 提示

- 範例中先以`range(1,82)`生成串列，總共包含81個元素，再以`reshape`產生第一個 $9 \times 9$ 的2維陣列，最後以`transpose`指令將陣列轉置，產生第二個 $9 \times 9$ 的2維陣列
- 本題所產生的陣列包含100個元素，範例中`range`使用的參數需要修改，以生成1-100的陣列元素
- 本題產生 $10 \times 10$ 的二維陣列，`reshape`指令中的參數需要進一步修改。

## Ex12B. (10 points) 使用range, array產生以下的一維陣列

```
array([ 1, 10, 19, 28, 37, 46, 55, 64, 73, 82, 91, 100, 109,
        118, 127, 136, 145, 154, 163, 172, 181, 190, 199, 208, 217, 226,
        235, 244, 253, 262, 271, 280, 289, 298, 307, 316, 325, 334, 343,
        352, 361, 370, 379, 388, 397, 406, 415, 424, 433, 442, 451, 460,
        469, 478, 487, 496, 505, 514, 523, 532, 541, 550, 559, 568, 577,
        586, 595, 604, 613, 622, 631, 640, 649, 658, 667, 676, 685, 694,
        703, 712, 721, 730, 739, 748, 757, 766, 775, 784, 793, 802, 811,
        820, 829, 838, 847, 856, 865, 874, 883, 892, 901, 910, 919, 928,
        937, 946, 955, 964, 973, 982, 991])
```

## Ex12C. (10 points) 使用巢狀迴圈計算 下列兩個矩陣的相加矩陣

```
[[ 1  2  3  4  5  6  7  8  9 10]
 [11 12 13 14 15 16 17 18 19 20]
 [21 22 23 24 25 26 27 28 29 30]
 [31 32 33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48 49 50]
 [51 52 53 54 55 56 57 58 59 60]
 [61 62 63 64 65 66 67 68 69 70]
 [71 72 73 74 75 76 77 78 79 80]
 [81 82 83 84 85 86 87 88 89 90]
 [91 92 93 94 95 96 97 98 99 100]]
```

```
[[ 1 11 21 31 41 51 61 71 81 91]
 [ 2 12 22 32 42 52 62 72 82 92]
 [ 3 13 23 33 43 53 63 73 83 93]
 [ 4 14 24 34 44 54 64 74 84 94]
 [ 5 15 25 35 45 55 65 75 85 95]
 [ 6 16 26 36 46 56 66 76 86 96]
 [ 7 17 27 37 47 57 67 77 87 97]
 [ 8 18 28 38 48 58 68 78 88 98]
 [ 9 19 29 39 49 59 69 79 89 99]
 [10 20 30 40 50 60 70 80 90 100]]
```

## Ex12D. (10 points) 產生兩個5x5的矩陣，使用concatenate合併為10x5的矩陣

```
array([[ 1,  2,  3,  4,  5],  
       [ 6,  7,  8,  9, 10],  
       [11, 12, 13, 14, 15],  
       [16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25]])
```

```
array([[ 1,  6, 11, 16, 21],  
       [ 2,  7, 12, 17, 22],  
       [ 3,  8, 13, 18, 23],  
       [ 4,  9, 14, 19, 24],  
       [ 5, 10, 15, 20, 25]])
```

```
array([[ 1,  2,  3,  4,  5],  
       [ 6,  7,  8,  9, 10],  
       [11, 12, 13, 14, 15],  
       [16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25],  
       [ 1,  6, 11, 16, 21],  
       [ 2,  7, 12, 17, 22],  
       [ 3,  8, 13, 18, 23],  
       [ 4,  9, 14, 19, 24],  
       [ 5, 10, 15, 20, 25]])
```

## Ex12E. (10 points) 產生兩個5x5的矩陣，使用hstack合併為5x10的矩陣

```
array([[ 1,  2,  3,  4,  5],  
       [ 6,  7,  8,  9, 10],  
       [11, 12, 13, 14, 15],  
       [16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25]])
```

```
array([[ 1,  6, 11, 16, 21],  
       [ 2,  7, 12, 17, 22],  
       [ 3,  8, 13, 18, 23],  
       [ 4,  9, 14, 19, 24],  
       [ 5, 10, 15, 20, 25]])
```

```
array([[ 1,  2,  3,  4,  5,  1,  6, 11, 16, 21],  
       [ 6,  7,  8,  9, 10,  2,  7, 12, 17, 22],  
       [11, 12, 13, 14, 15,  3,  8, 13, 18, 23],  
       [16, 17, 18, 19, 20,  4,  9, 14, 19, 24],  
       [21, 22, 23, 24, 25,  5, 10, 15, 20, 25]])
```

**Ex12F. (10 points)** 使用`range`, `reshape`產生以下的二維陣列的兩個子矩陣，以及這兩個子矩陣的合併矩陣

```
[[ 1  2  3  4  5  6  7  8  9 10]
 [11 12 13 14 15 16 17 18 19 20]
 [21 22 23 24 25 26 27 28 29 30]
 [31 32 33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48 49 50]
 [51 52 53 54 55 56 57 58 59 60]
 [61 62 63 64 65 66 67 68 69 70]
 [71 72 73 74 75 76 77 78 79 80]
 [81 82 83 84 85 86 87 88 89 90]
 [91 92 93 94 95 96 97 98 99 100]]
```

```
array([[ 11,  12,  13,  14,  15,  16,  18,  19,  20],
       [ 21,  22,  23,  24,  25,  26,  28,  29,  30],
       [ 31,  32,  33,  34,  35,  36,  38,  39,  40],
       [ 41,  42,  43,  44,  45,  46,  48,  49,  50],
       [ 51,  52,  53,  54,  55,  56,  58,  59,  60],
       [ 61,  62,  63,  64,  65,  66,  68,  69,  70],
       [ 71,  72,  73,  74,  75,  76,  78,  79,  80],
       [ 81,  82,  83,  84,  85,  86,  88,  89,  90],
       [ 91,  92,  93,  94,  95,  96,  98,  99, 100]])
```

## Ex12G. (10 points) 產生以下的三維陣列

```
array([[ [ 1,  2,  3],  
        [ 4,  5,  6],  
        [ 7,  8,  9]],  
       [[10, 11, 12],  
        [13, 14, 15],  
        [16, 17, 18]],  
       [[19, 20, 21],  
        [22, 23, 24],  
        [25, 26, 27]])
```