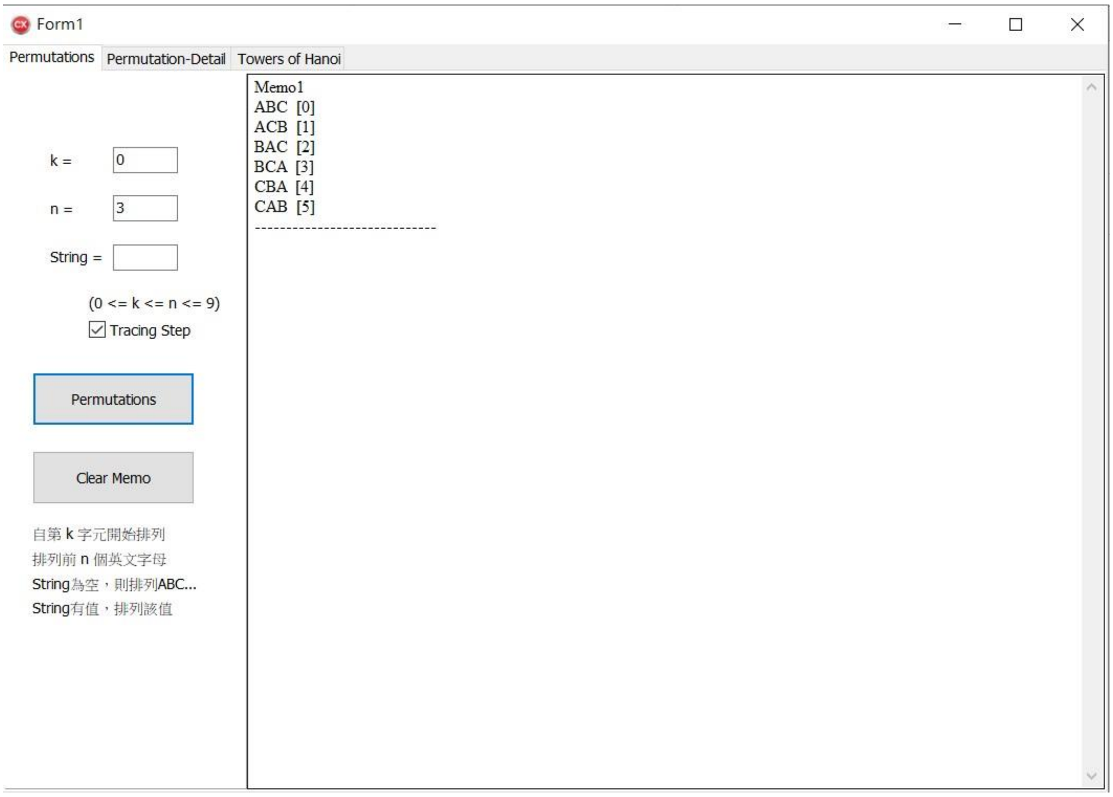
HW2 Permutations and Hanoi Towers

0616098 黃秉茂

執行結果：

Permutation：

k = 0, n = 3, with tracing step



k = 0

n = 3

String =

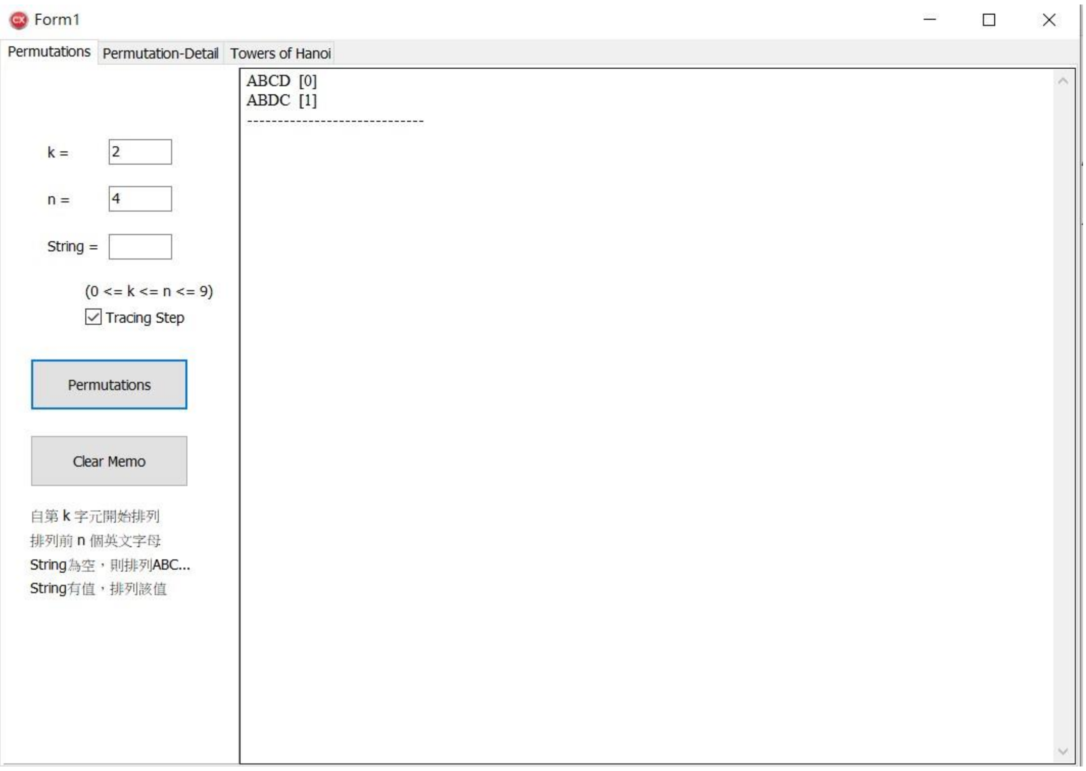
(0 <= k <= n <= 9)

☑ Tracing Step

```
Memo1
ABC  [0]
ACB  [1]
BAC  [2]
BCA  [3]
CBA  [4]
CAB  [5]
------------------------------
```

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

k = 2, n = 4, with tracing step



k = 2

n = 4

String =

(0 <= k <= n <= 9)

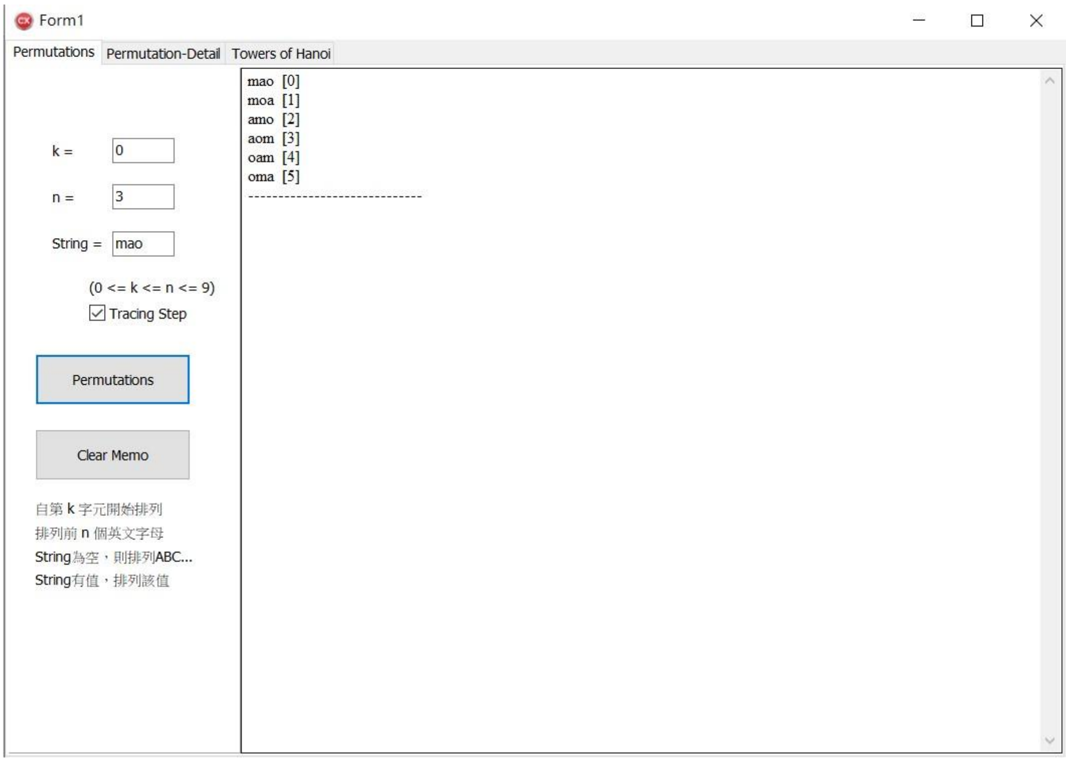
☑ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
ABCD  [0]
ABDC  [1]
-----------------------------
```

k = 0, n = 3, String = “mao” with tracing detail, permutation

String(= winni) with first n(= 3) character with tracing detail



k = 0

n = 3

String = mao

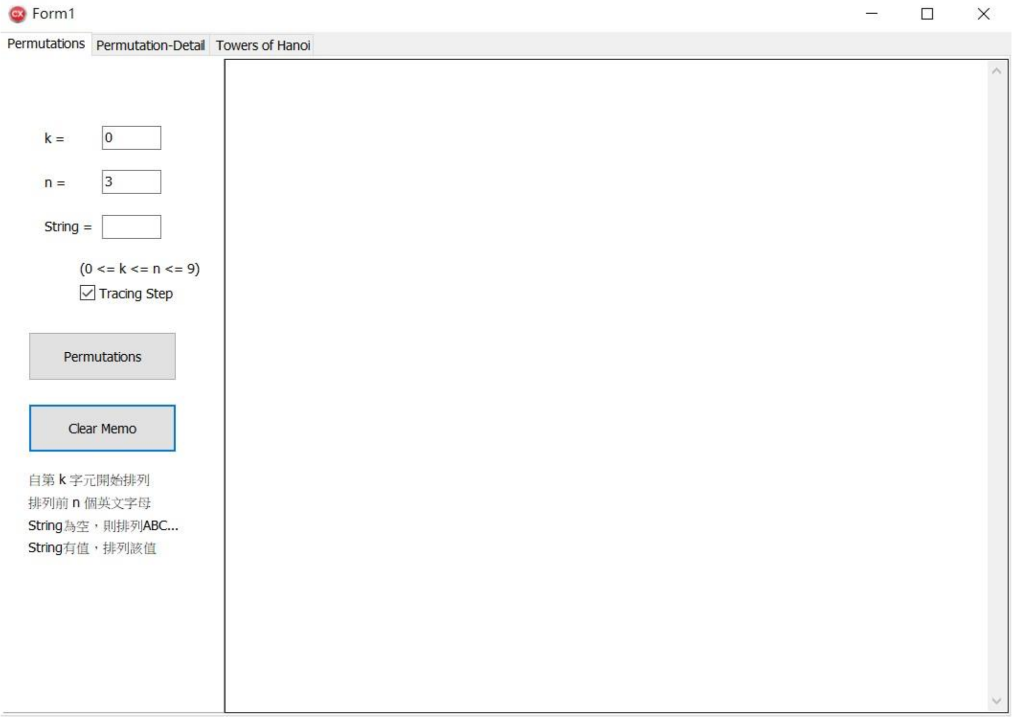
(0 <= k <= n <= 9)

☑ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
mao  [0]
moa  [1]
amo  [2]
aom  [3]
oam  [4]
oma  [5]
-----------------------------
```

Clear memo



k = 0

n = 3

String =

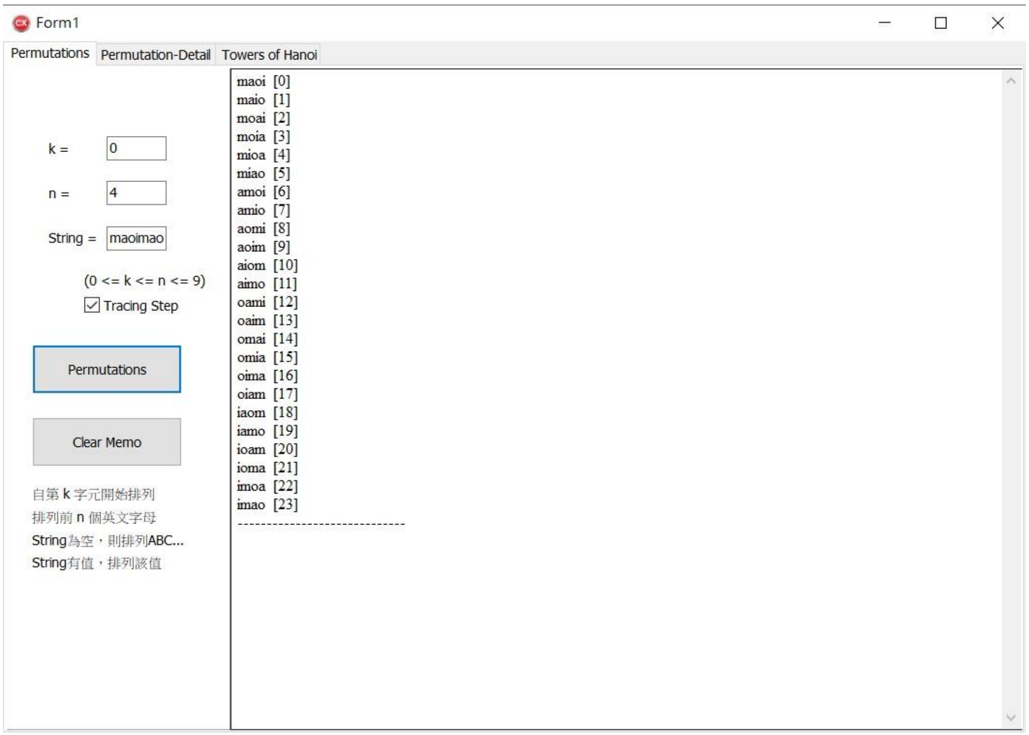
(0 <= k <= n <= 9)

☑ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

k = 0, n = 4, String = “maoimao” with tracing detail,

permutation String(= maoimao) with first n(= 4) character with tracing detail



k = 0

n = 4

String = maoimao

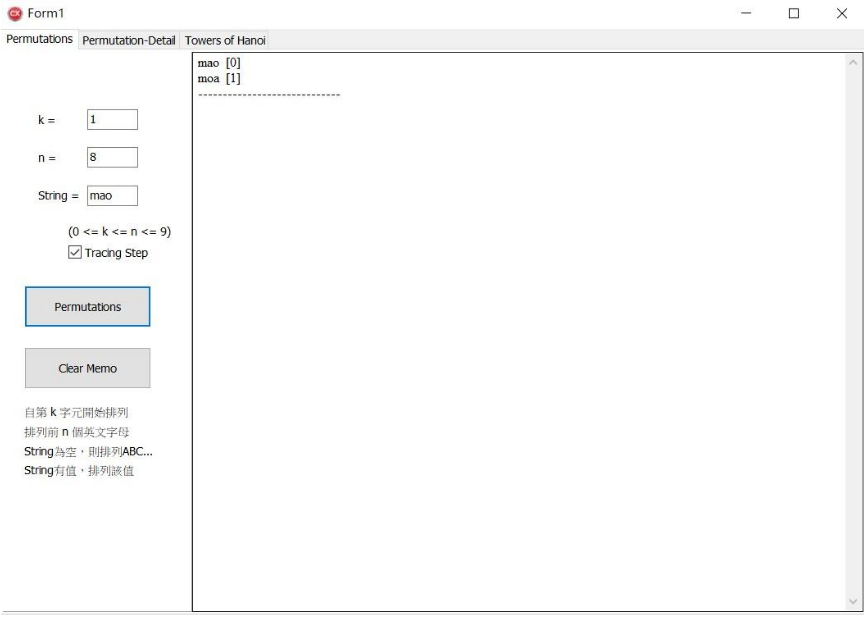
(0 <= k <= n <= 9)

☑ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
maoi  [0]
maio  [1]
moai  [2]
moia  [3]
mioa  [4]
miao  [5]
amoi  [6]
amio  [7]
aomi  [8]
aoim  [9]
aiom  [10]
aimo  [11]
oami  [12]
oaim  [13]
omai  [14]
omia  [15]
oima  [16]
oiam  [17]
iaom  [18]
iamo  [19]
ioam  [20]
ioma  [21]
imoa  [22]
imao  [23]
-----------------------------
```

k = 1, n = 8, String = “mao” with tracing detail since n > length of the String, permutation String(= mao) with tracing detail



k = 1

n = 8

String = mao

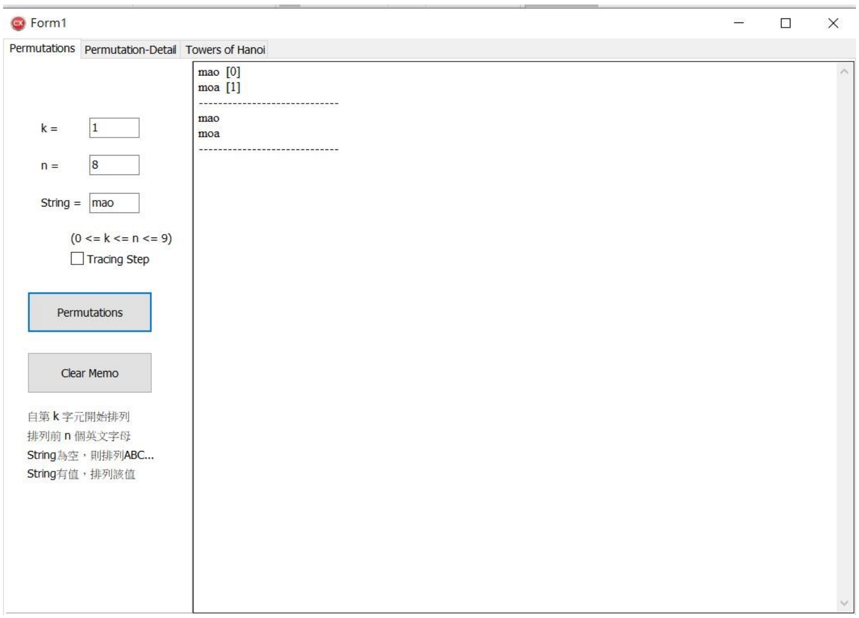
(0 <= k <= n <= 9)

☑ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
mao  [0]
moa  [1]
-----------------------------
```

k = 1, n = 8, String = “mao” without tracing detail since n > length of the String, permutation String(= mao) without tracing detail



k = 1

n = 8

String = mao

(0 <= k <= n <= 9)

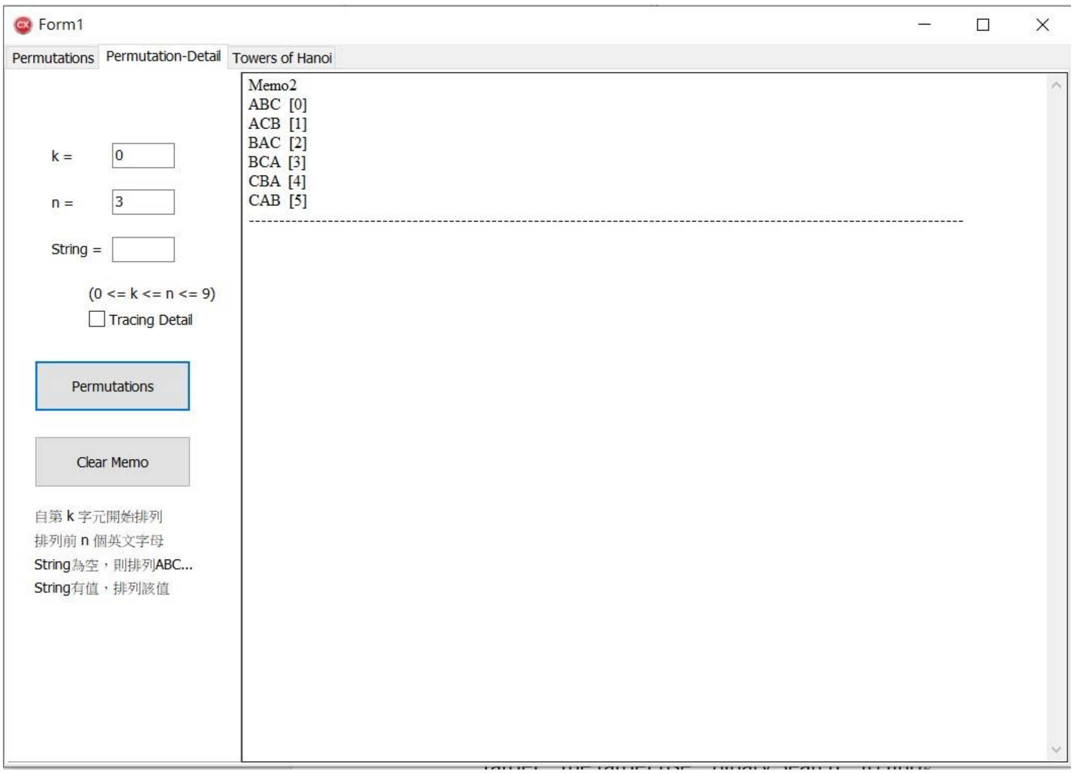
□ Tracing Step

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
mao  [0]
moa  [1]
-----------------------------
mao
moa
-----------------------------
```

Permutation Detail：

k = 0, n = 3, without tracing detail



k = 0

n = 3

String =

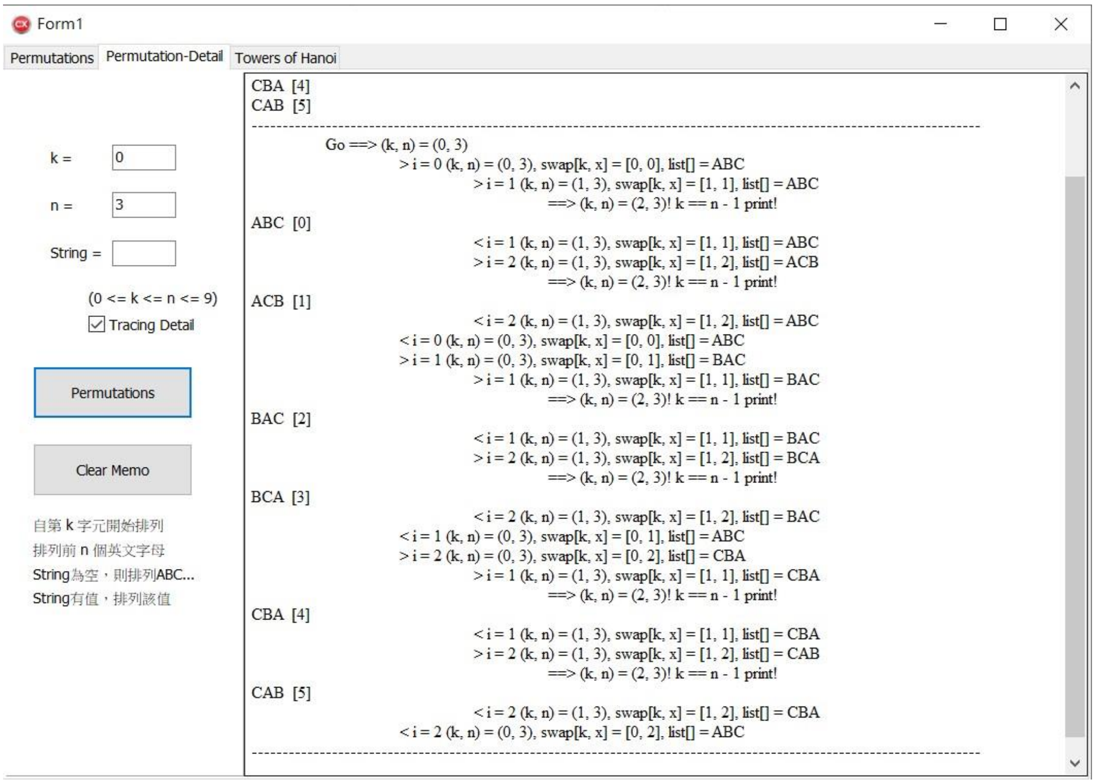
(0 <= k <= n <= 9)

□ Tracing Detail

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
Memo2
ABC  [0]
ACB  [1]
BAC  [2]
BCA  [3]
CBA  [4]
CAB  [5]
--------------------------------------------------------------------------------------------------------
```

k = 0, n = 3, with tracing detail



k = 0

n = 3

String =

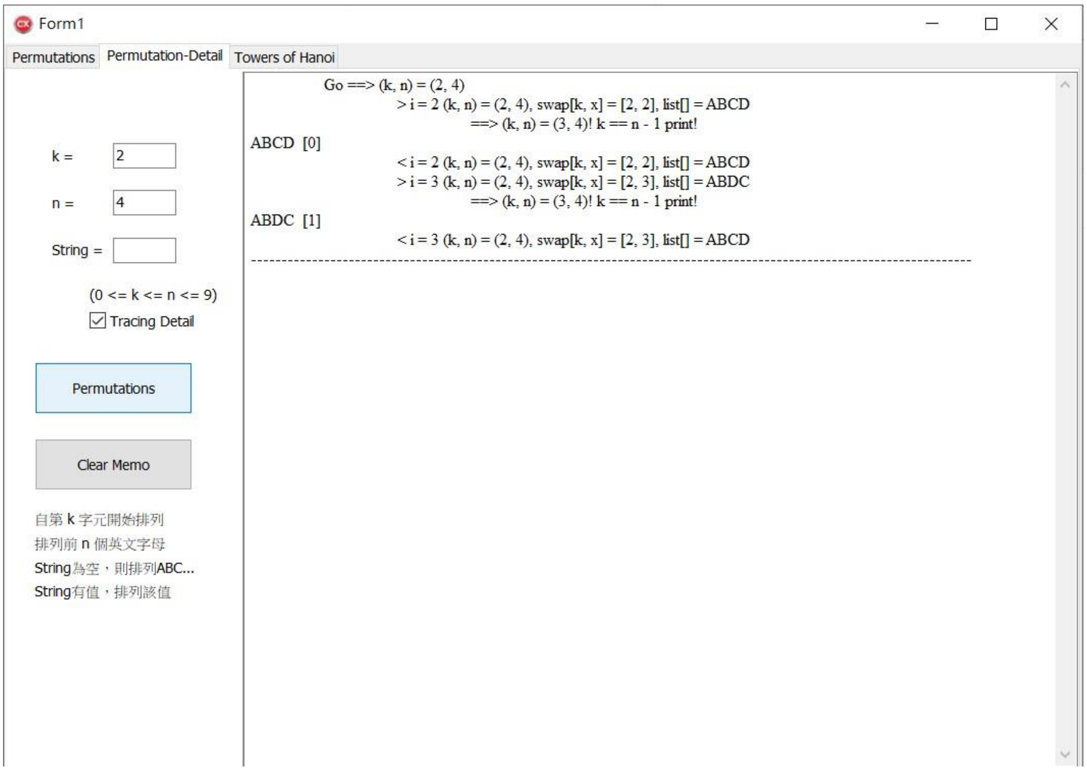
(0 <= k <= n <= 9)

☑ Tracing Detail

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
CBA  [4]
CAB  [5]
--------------------------------------------------------------------------------------------------------
          Go ==> (k, n) = (0, 3)
                    > i = 0 (k, n) = (0, 3), swap[k, x] = [0, 0], list[] = ABC
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = ABC
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
ABC  [0]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = ABC
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = ACB
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
ACB  [1]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = ABC
                    < i = 0 (k, n) = (0, 3), swap[k, x] = [0, 0], list[] = ABC
                    > i = 1 (k, n) = (0, 3), swap[k, x] = [0, 1], list[] = BAC
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = BAC
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
BAC  [2]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = BAC
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = BCA
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
BCA  [3]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = BAC
                    < i = 1 (k, n) = (0, 3), swap[k, x] = [0, 1], list[] = ABC
                    > i = 2 (k, n) = (0, 3), swap[k, x] = [0, 2], list[] = CBA
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = CBA
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
CBA  [4]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = CBA
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = CAB
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
CAB  [5]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = CBA
                    < i = 2 (k, n) = (0, 3), swap[k, x] = [0, 2], list[] = ABC
--------------------------------------------------------------------------------------------------------
```

k = 2, n = 4, with tracing detail



k = 2

n = 4

String =

(0 <= k <= n <= 9)

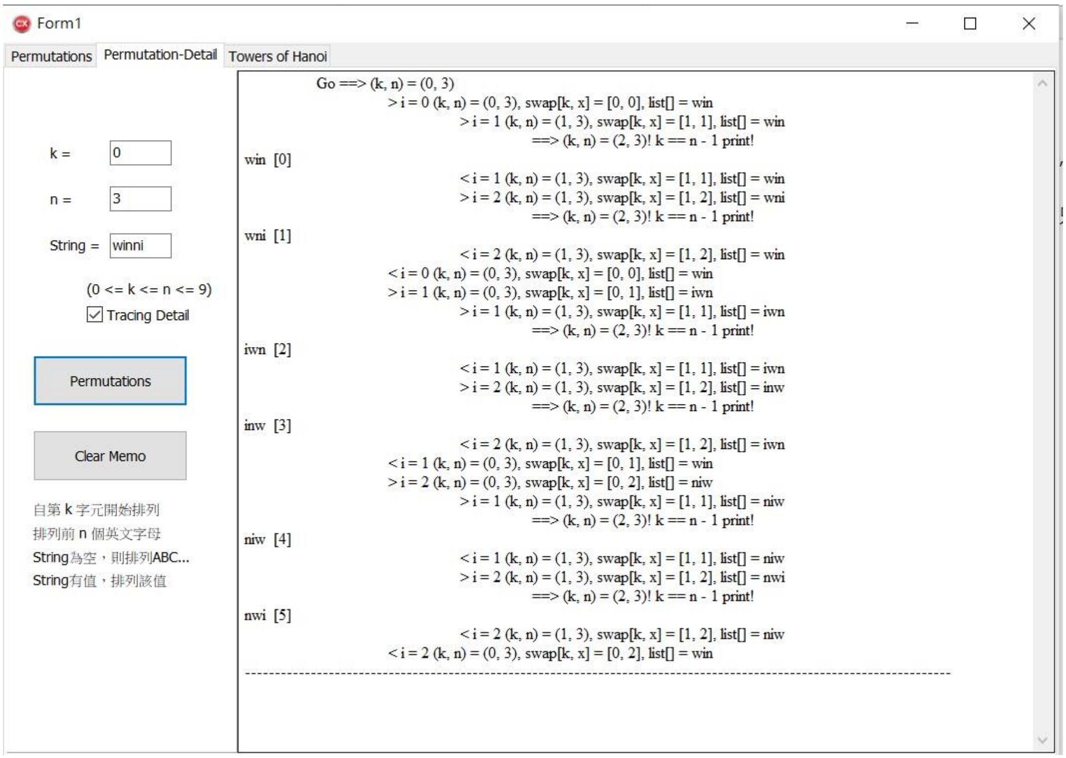
☑ Tracing Detail

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
            Go ==> (k, n) = (2, 4)
                    > i = 2 (k, n) = (2, 4), swap[k, x] = [2, 2], list[] = ABCD
                              ==> (k, n) = (3, 4)! k == n - 1 print!
ABCD  [0]
                    < i = 2 (k, n) = (2, 4), swap[k, x] = [2, 2], list[] = ABCD
                    > i = 3 (k, n) = (2, 4), swap[k, x] = [2, 3], list[] = ABDC
                              ==> (k, n) = (3, 4)! k == n - 1 print!
ABDC  [1]
                    < i = 3 (k, n) = (2, 4), swap[k, x] = [2, 3], list[] = ABCD
--------------------------------------------------------------------------------------------------------
```

k = 0, n = 3, String = “winni” with tracing detail, permutation

String(= winni) with first n(= 3) character with tracing detail



k = 0

n = 3

String = winni

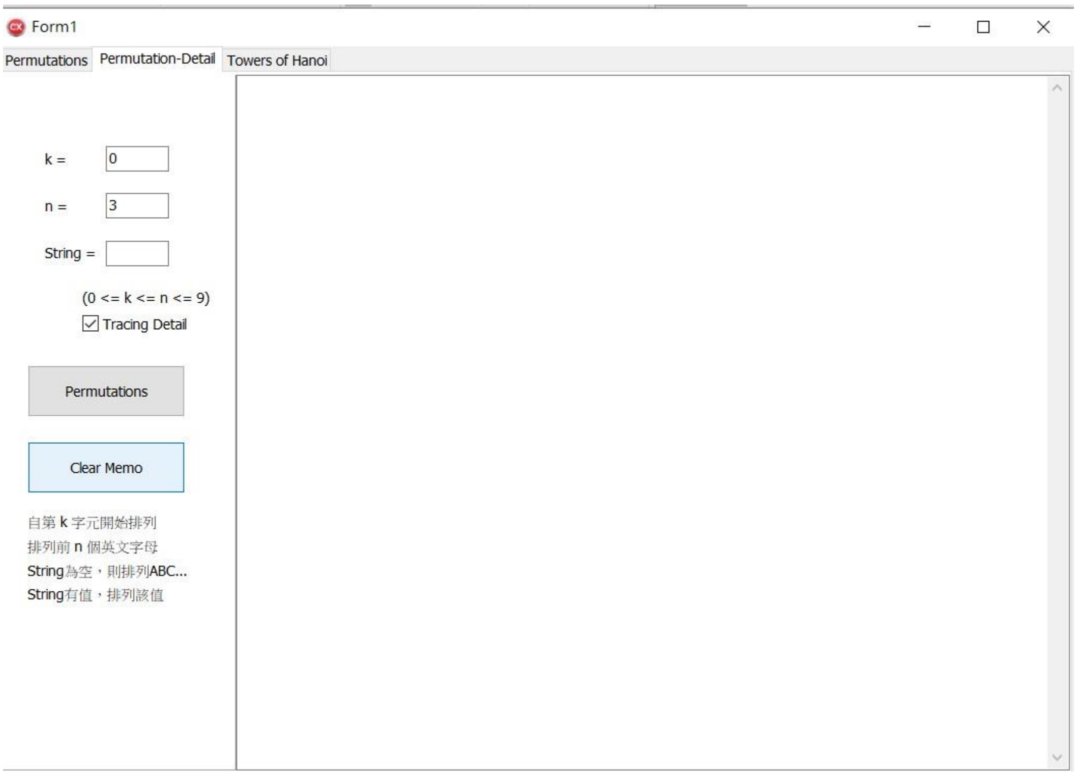
(0 <= k <= n <= 9)

☑ Tracing Detail

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

```
            Go ==> (k, n) = (0, 3)
                    > i = 0 (k, n) = (0, 3), swap[k, x] = [0, 0], list[] = win
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = win
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
win  [0]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = win
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = wni
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
wni  [1]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = win
                    < i = 0 (k, n) = (0, 3), swap[k, x] = [0, 0], list[] = win
                    > i = 1 (k, n) = (0, 3), swap[k, x] = [0, 1], list[] = iwn
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = iwn
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
iwn  [2]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = iwn
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = inw
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
inw  [3]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = iwn
                    < i = 1 (k, n) = (0, 3), swap[k, x] = [0, 1], list[] = win
                    > i = 2 (k, n) = (0, 3), swap[k, x] = [0, 2], list[] = niw
                              > i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = niw
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
niw  [4]
                              < i = 1 (k, n) = (1, 3), swap[k, x] = [1, 1], list[] = niw
                              > i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = nwi
                                        ==> (k, n) = (2, 3)! k == n - 1 print!
nwi  [5]
                              < i = 2 (k, n) = (1, 3), swap[k, x] = [1, 2], list[] = niw
                    < i = 2 (k, n) = (0, 3), swap[k, x] = [0, 2], list[] = win
--------------------------------------------------------------------------------------------------------
```

Clear memo



k = 0

n = 3

String =

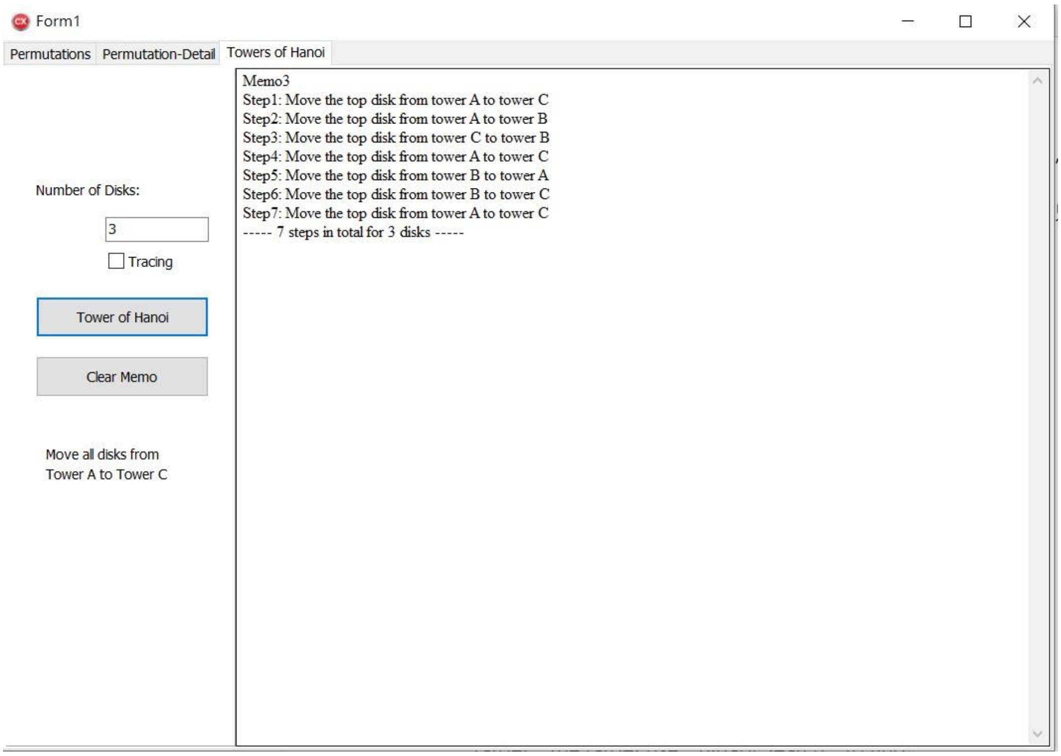
(0 <= k <= n <= 9)

☑ Tracing Detail

自第 k 字元開始排列
排列前 n 個英文字母
String為空，則排列ABC...
String有值，排列該值

Towers of Hanoi：

n=3, without tracing



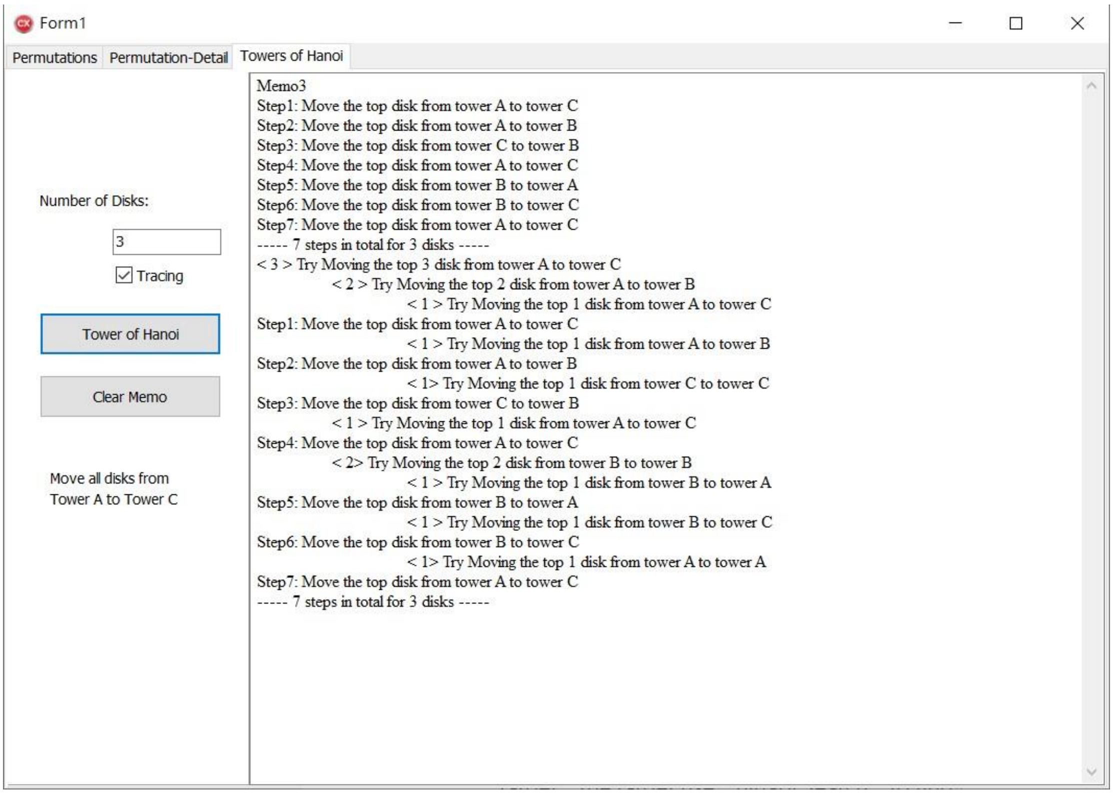
Number of Disks: 3

□ Tracing

Move all disks from
Tower A to Tower C

```
Memo3
Step1: Move the top disk from tower A to tower C
Step2: Move the top disk from tower A to tower B
Step3: Move the top disk from tower C to tower B
Step4: Move the top disk from tower A to tower C
Step5: Move the top disk from tower B to tower A
Step6: Move the top disk from tower B to tower C
Step7: Move the top disk from tower A to tower C
----- 7 steps in total for 3 disks -----
```

n=3, with tracing



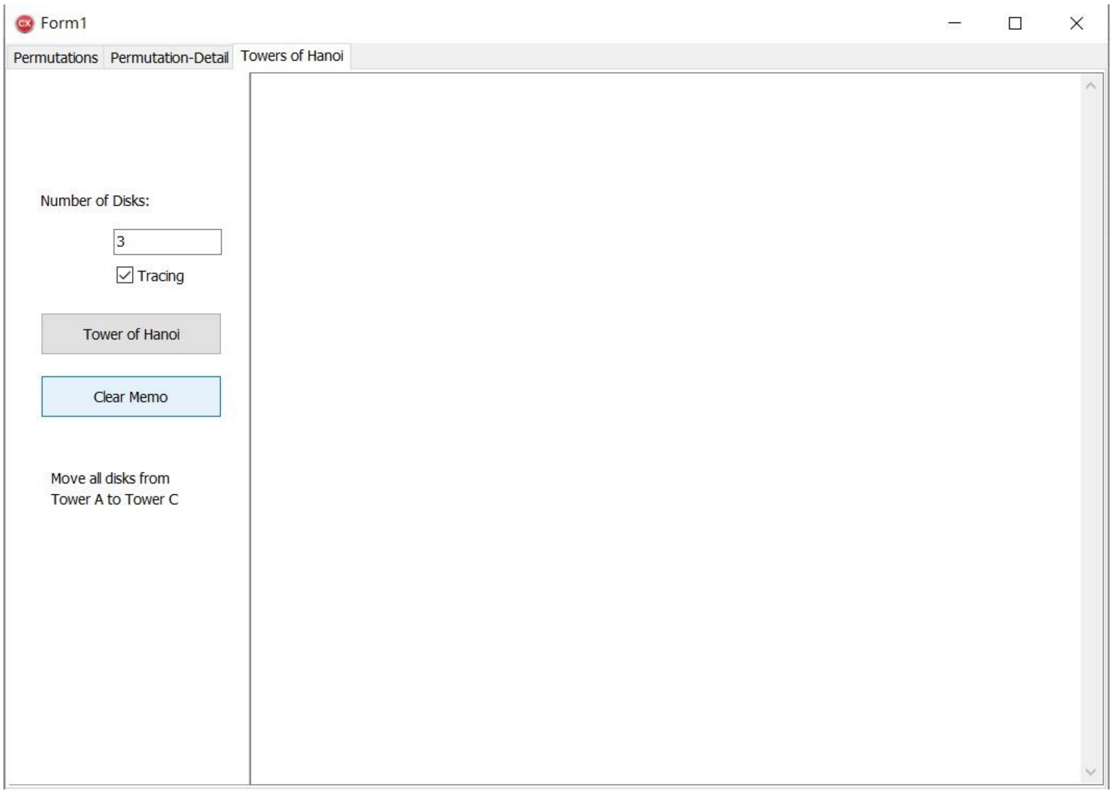
Number of Disks: 3

☑ Tracing

Move all disks from Tower A to Tower C

```
Memo3
Step1: Move the top disk from tower A to tower C
Step2: Move the top disk from tower A to tower B
Step3: Move the top disk from tower C to tower B
Step4: Move the top disk from tower A to tower C
Step5: Move the top disk from tower B to tower A
Step6: Move the top disk from tower B to tower C
Step7: Move the top disk from tower A to tower C
----- 7 steps in total for 3 disks -----
< 3 > Try Moving the top 3 disk from tower A to tower C
          < 2 > Try Moving the top 2 disk from tower A to tower B
                    < 1 > Try Moving the top 1 disk from tower A to tower C
Step1: Move the top disk from tower A to tower C
                    < 1 > Try Moving the top 1 disk from tower A to tower B
Step2: Move the top disk from tower A to tower B
                    < 1> Try Moving the top 1 disk from tower C to tower C
Step3: Move the top disk from tower C to tower B
          < 1 > Try Moving the top 1 disk from tower A to tower C
Step4: Move the top disk from tower A to tower C
          < 2> Try Moving the top 2 disk from tower B to tower B
                    < 1 > Try Moving the top 1 disk from tower B to tower A
Step5: Move the top disk from tower B to tower A
                    < 1 > Try Moving the top 1 disk from tower B to tower C
Step6: Move the top disk from tower B to tower C
                    < 1> Try Moving the top 1 disk from tower A to tower A
Step7: Move the top disk from tower A to tower C
----- 7 steps in total for 3 disks -----
```

Clear memo



Number of Disks: 3

☑ Tracing

Move all disks from Tower A to Tower C

程式說明：

Permutation：

Edit：

k：自第 k 字元開始排列

n：排列前 n 個英文字母

String：String 為空，則排列 ABC...，String 有值，排列 String 的該值

String 有值時，當 n = 0 or n > length of String，都是排列整串該 String

Check-box：

Tracing Step：要不要把所有排列的順序印出來

Button：

Permutations：排列

Clear Memo：清除 memo

Permutation Detail：

Edit：

k：自第 k 字元開始排列

n：排列前 n 個英文字母

String：String 為空，則排列 ABC...，String 有值，排列 String 的該值

String 有值時，當 n = 0 or n > length of String，都是排列整串該 String

Check-box：

Tracing Detail：要不要把所有排列的交換步驟印出來

Button：

Permutations：排列

Clear Memo：清除 memo

Towers of Hanoi：

Edit：

Number of Disks：要疊幾片

Check-box：

Tracing：要不要印出更細節的部份

Button：

Tower of Hanoi：執行 Towers of Hanoi

Clear Memo：清除 memo

主要是利用 recursive 完成此次作業