

Unit 13.

List ສອງມິຕີ (Two-Dimensional Lists)

Learning objectives

- ✓ ສາມາດສ້າງ list ສອງມືຕີ ແລະ ເຂົ້າເຖິງອີງປະກອບຂອງມັນໂດຍນໍາໃຊ້ຕຳແໜ່ງຂໍ້ມູນ.
- ✓ ສາມາດກຳນົດຄ່າໃຫ້ກັບອີງປະກອບຂອງ list ແລະ ສະແດງຜົນ.
- ✓ ເຂົ້າໃຈໃນການສ້າງ array ສອງມືຕີ, ກຳນົດຄ່າ ພ້ອມທັງສາມາດ ສ້າງ list ສອງມືຕີທີ່ຊ້າວັນ ໂດຍນໍາໃຊ້ຕຳສັ່ງ loop.
- ✓ ເຂົ້າໃຈຄຸນສົມບັດຂອງ list ສອງມືຕີໃນ Python ແລະ ສ້າງ jagged lists.

Learning overview

- ✓ ສ້າງ list ສອງມືຕີ ແລະ ກຳນົດຄ່າໃຫ້ກັບອົງປະກອບຂອງມັນໂດຍນໍາໃຊ້ຕຳແໜ່ງຂຶ້ນ ຫຼື index.
- ✓ ພິມຄ່າໃນ list ສອງມືຕີ ໂດຍນໍາໃຊ້ຄໍາສ້າງ loop
- ✓ ສ້າງ jagged list ໂດຍນໍາໃຊ້ຄໍາສ້າງ loop
- ✓ ນໍາໃຊ້ contraction expressions ເພື່ອກຳນົດ list ສອງມືຕີ

Concepts You Will Need to Know From Previous Units

- ✓ ການສ້າງ ແລະ ການເຊື່ອມຕໍ່ sequence objects
- ✓ ການເຂົ້າເຖິງອົງປະກອບໂດຍນໍາໃຊ້ຕຳແໜ່ງຂອງ sequence object
- ✓ ນໍາໃຊ້ຄໍາສ້າງ double loop ແລະ key methods ຂອງ sequence objects

Keywords

**Two-Dimensional
Lists**

**Two-Dimensional
Indexing**

Jagged Lists

**Double Loop
Statement**

Mission

1. Real world problem

1.1. ການຖອຍລົງຂອງອຸດສາຫະກຳຮູບເງິນ ເນື້ອງຈາກການພັດທະນາຂອງ OTT



- ▶ ประจุบัน, เมื่อวิจารณ์การพัฒนาทางด้านตัวโน้ลฉี บลีกานสีแบบ over-the-top(OTT) เช่น: Netflix และ YouTube เร็กดให้อุดสหะทำรูบเงินทำลังปะสิบกับวิกิตกานถึงไทย.
 - ▶ อุดสหะทำรูบเงินมีส่วนสำคัญต่อเสนอภารกิจท้องที่นั่น สะมัน ท้าโรงรูบเงินท้องที่นั่นปิด อัคตากานว่างงานของคืนกำจะเพิ่มขึ้น.
 - ▶ ประจุบัน โรงรูบเงินทำลังเงินวิชานะวิกิตกาน ด้วยกานพัฒนาภานบลีกาน ลูกถ้า และ กานบลีกานเพิ่มเติมปีๆ นี่.
 - ▶ ส่วนนี่ของกานบลีกานนี้, โรงรูบเงินจะน้อยบางแข็งพ้อมบลีกาน ละดับ premium เช่นว่า ตั้งปีบ่อนละดับ high-end.

1. Real world problem

1.2. ລະບົບຈອງບ່ອນນັ່ງໃນໂຮງຮູບເງິ



- ▶ ໃນ mission ນີ້, ຈະສ້າງລະບົບຈອງບ່ອນນັ່ງໃນໂຮງຮູບເງິ
- ▶ ປະຈຸບັນ ໂຮງຮູບເງິມີລະບົບຈອງບ່ອນນັ່ງ ແລະ ສາມາດກວດສອບບ່ອນນັ່ງໃນ smartphone ຫຼື counter ໃນໂຮງຮູບເງິໄດ້.
- ▶ ຕອນນີ້ ຈະສ້າງລະບົບແຈ້ງເຕືອນບ່ອນນັ່ງໃນໂຮງຮູບເງິຂະໜາດນີ້ຍີ້ມີ 18 ບ່ອນນັ່ງ, 3 ແຖວ ແລະ 6 ທັນ ດັ່ງສະແດງໃນຮູບ.
- ▶ ໃນລະບົບນີ້ ຄວນມີຟັງຊັນສ້າງບ່ອນນັ່ງຫວ່າງ ແລະ ບ່ອນນັ່ງສໍາຮອງແບບສຸ່ມ ແລະ ບອກຈຳນວນບ່ອນນັ່ງຫວ່າງ.

1. Real world problem

1.3. ຕົວຢ່າງລະບົບຈອງບ່ອນນັ້ງໃນໂຮງຮູບເງິ

The number of empty seats : 5

- ສໍາລັບການຈໍາລອງ, ໃຫ້ສ້າງ list ສອງມີຕີ ແລະ ກຳນົດໃຫ້ເລກ 0 ເປັນບ່ອນນັ້ງທວ່າງ ແລະ ເລກ 1 ເປັນບ່ອນນັ້ງບໍ່ທວ່າງ.
- ບ່ອນນັ້ງໃນໂຮງຮູບເງິນີ້ ປະກອບເວຼຍ list ສອງມີຕີ ເປັນໂຮງຮູບເງິຂະໜາດນັ້ນ ແລະ ສົມມຸດວ່າມີບ່ອນນັ້ງ 18 ບ່ອນ ໃນນີ້ ມີ 3 ແຖວ ແລະ 6 ຖັນ.
- ບ່ອນນັ້ງທວ່າງ ແລະ ບ່ອນນັ້ງທີ່ບໍ່ທວ່າງ ຈະຖືກສຸມຂຶ້ນມາ ໂດຍຜ່ານ random module ເຊິ່ງບ່ອນນັ້ງຈະກຳນົດຄ່າເປັນ 1 ຫຼື 0.
- ເລີ່ມສ້າງໂປຣແກຣມ ເພື່ອພິມຈໍານວນບ່ອນນັ້ງທວ່າງ

1. Real world problem

1.4. ຂໍ້ມູນອ້າງອີງເພີ່ມຕົວສໍາລັບການຈອງບ່ອນນັ່ງໃນໂຮງຮຽນ



2. Mission

2.1. ការនេះតាមរបៀបខាងក្រោម

```
1 import random
2
3 seat = []
4
5 for i in range(3):      # Randomly input occupied/unoccupied seats
6     line = []
7     for j in range(6):
8         rand = random.randrange(0, 2) # Use random numbers of 0 and 1
9         line.append(rand)
10    seat.append(line)
11
12 available = 0
13
14 for i in range(3):      # Print the seats, and calculate the number of remaining seats
15     for j in range(6):
16         print(seat[i][j], ' ', end='')
17         if seat[i][j] == 0:
18             available += 1
19     print()
20 print('The number of remaining seats in the theater:', available)
```

0 0 1 0 1 1
0 1 1 0 1 1
1 1 1 1 1 0

The number of remaining seats in the theater: 6

2. Mission

2.1. ການເຮັດວຽກຂອງລະບົບຈອງບ່ອນນັ່ງໃນໂຮງຮູບເງິ

| ເນື່ອງຈາກ code ດຳສັ່ງຂ້ອນຂ້າງຍາວ ດັ່ງນັ້ນຈຶ່ງແບ່ງ flowchart ອອກເປັນ ສອງສ່ວນ ດັ່ງສະແດງລຸ່ມນີ້.

- ▶ Flowchart 1: ເປັນ Algorithm ທີ່ສຸມບ່ອນນັ່ງຫວ່າງໃນໂຮງຮູບເງິ
- ▶ Flowchart 2: ເປັນ Algorithm ທີ່ຄົ້ນຫາຈຳນວນບ່ອນນັ່ງຫວ່າງທັງໝົດໃນໂຮງຮູບເງິ

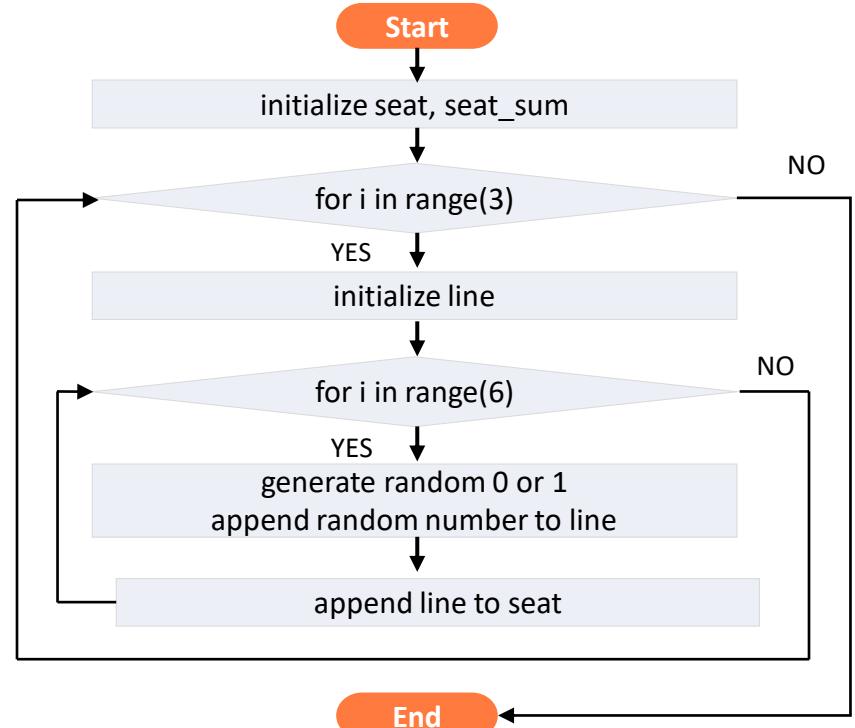
2. Mission

2.2. Programming Plan

Pseudocode 1: Generating random seats

```
[1] Start  
[2] Reset the seat list.  
[3] for i = 0 for three times do {  
    [4] Reset the line list.  
    [5] for j = 0 for six times do{  
        [6] Generate random numbers between 0 and 1.  
        Add those random numbers to the line list.  
        [7] Add line to the seat list. }  
    }  
[8] End
```

Flowchart 1: Generating random seats



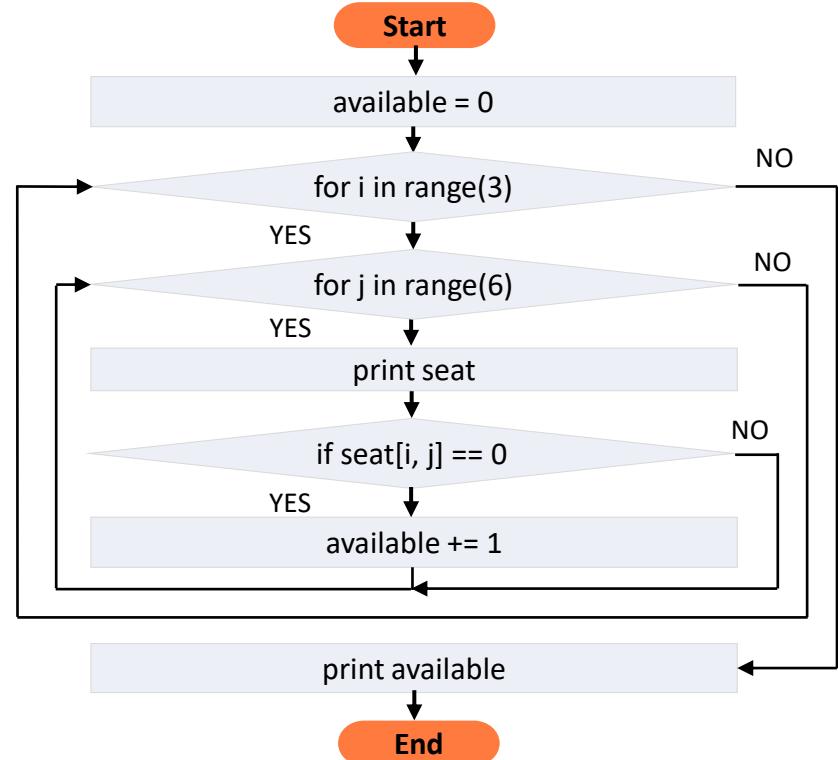
2. Mission

2.2. Programming Plan

Pseudocode 2: Printing seat information

```
[1] Start  
[2] Reset the number of empty seats to 0.  
[3] for i = 0 to for three times do {  
    [4] for j = 0 to for six times do{  
        [5] Print seat[i, j].  
        [6] if seat[i, j] == 0 {  
            [7] Increase the number of empty seats by 1.  
        }  
    }  
    [8] Print the number of empty seats.  
[9] End
```

Flowchart 2 : Printing seat information



2. Mission

2.3. Final code ຂອງລະບົບຈອງປ່ອນນັ້ງໃນໂຄງຮູບເງິນ

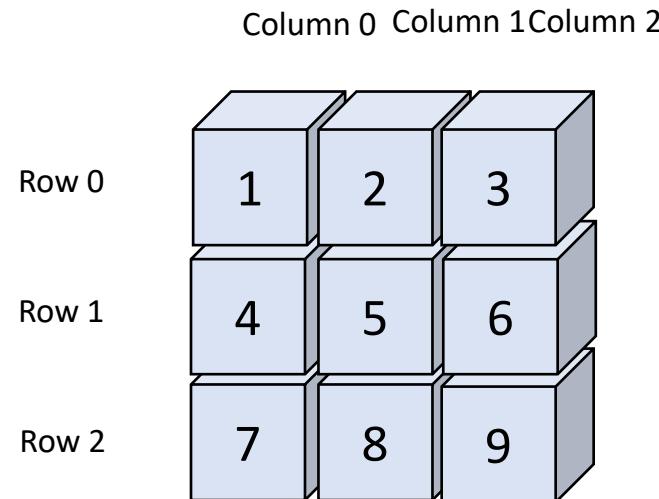
```
1 import random
2
3 seat = []
4
5 for i in range(3):      # Randomly input occupied/unoccupied seats
6     line = []
7     for j in range(6):
8         rand = random.randrange(0, 2)  # Use random numbers of 0 and 1
9         line.append(rand)
10    seat.append(line)
11
12 available = 0
13
14 for i in range(3):      # Print the seat, and calculate the number of remaining seats
15     for j in range(6):
16         print(seat[i][j], ' ', end=' ')
17         if seat[i][j] == 0:
18             available += 1
19     print()
20 print('The number of remaining seats in the theater:', available)
```

| Key concept

1. ສ້າງ List ສອງມືຕີ

1.1. List ສອງມືຕີ ທີ່ມີ 3 ແຖວ ແລະ 3 ຫັນ

- | List ສອງມືຕີ ມີໄດ້ສ້າງທີ່ມີການຈັດກຸ່ມເປັນ list ຫຼືມືຕີ ແລະ ເປັນທີ່ນີ້ຍົມໃຊ້ກັນຢ່າງກວ້າງຂວາງ.
- | ການສ້າງ list ສອງມືຕີ ໝາຍເຖິງການສ້າງ list ຊ້ອນເຂົ້າໃນ list.
- | ຈາກຮູບລຸ່ມນີ້ ແມ່ນການສ້າງ list ສອງມືຕີທີ່ປະກອບດ້ວຍ 3 ແຖວ ແລະ 3 ຫັນ.



1. ສ້າງ List ສອງມືຕີ

1.1. ຕົວຢ່າງການສ້າງ list ສອງມືຕີທີ່ປະກອບດ້ວຍ 3 ແຖວ ແລະ 3 ຖັນ

| ສ້າງ list ສອງມືຕີ ທີ່ບັນຈຸອີງປະກອບໃນ list.

```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

- ▶ ນີ້ແມ່ນ list ສອງມືຕີທີ່ມີຂະໜາດ 3×3 ເຊິ່ງປະກອບເປັນກຸ່ມ list ສາມກຸ່ມ ແຕ່ລະ ກຸ່ມມີ 3 ຕົວເລກ.

```
1 list_array = [[1, 2, 3],  
2                 [4, 5, 6],  
3                 [7, 8, 9]]
```

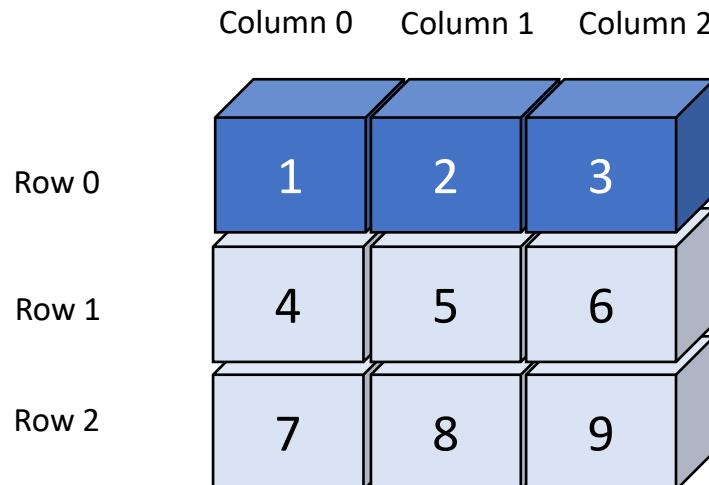
- ▶ ເພື່ອໃຫ້ງ່າຍໃນການຈໍາແນກ list ສອງມືຕີ, ບ້ອນຄ່າເປັນ ສາມແຖວຕາມລວງນອນ ແລະ ສາມ ແຖວຕາມລວງຕັ້ງ.
- ▶ ສອງວິທີນີ້ຄືກັນ ແຕ່ວິທີທີ່ສອງຈະເຂົ້າໃຈ່ງ່າຍກວ່າ.

	Column 0	Column 1	Column 2
Row 0	1	2	3
Row 1	4	5	6
Row 2	7	8	9

1. ສ້າງ List ສອງມືຕີ

1.1. ຕົວຢ່າງການສ້າງ list ສອງມືຕີທີ່ປະກອບດ້ວຍ 3 ແຖວ ແລະ 3 ຖັນ

| ສ້າງ list ສອງມືຕີທີ່ມີອີງປະກອບໃນ list ດັ່ງຮູບລຸ່ມນີ້. list_array[0] ຈະອ້າງອີງອີງປະກອບທຳອິດຄື [1, 2, 3].



```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
2 list_array[0]
```

```
[1, 2, 3]
```

2. ការເຂົ້າເຖິງອີງປະກອບຂອງ List ສອງມືຕີ

2.1. ຕໍາແໜ່ງການເຂົ້າເຖິງອີງປະກອບໃນ list

| ການເຂົ້າເຖິງອີງປະກອບຂອງ list ສອງມືຕີ ໂດຍໃຊ້ ຕໍາແໜ່ງຂຶ້ນ (index).

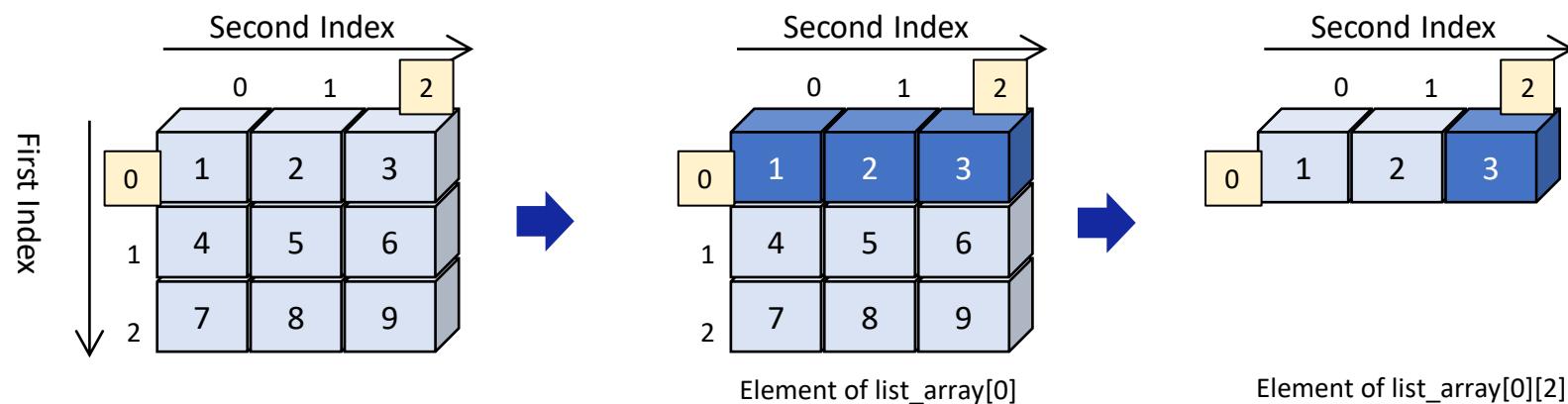
```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 list_array[0]
```

[1, 2, 3]

```
1 list_array[0][2]
```

3

- ▶ ການເຂົ້າເຖິງຕໍາແໜ່ງອີງປະກອບໃນ list ສອງມືຕີ ແມ່ນນຳໃຊ້ເຄື່ອງໝາຍ [] ສອງເຄື່ອງໝາຍ, ໃນນີ້ເຄື່ອງໝາຍທຳອິດ ແມ່ນຕໍາແໜ່ງແຖວ ແລະ ເຄື່ອງໝາຍທີ່ສອງ ແມ່ນຕໍາແໜ່ງທັນ.



2. ການເຂົ້າເຖິງອົງປະກອບຂອງ List ສອງມືຕີ

2.2. ການເຂົ້າຄ່າຂອງອົງປະກອບໃນ list ສອງມືຕີ ໂດຍນຳໃຊ້ຕຳແໜ່ງຂໍ້ມູນ

```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

- ▶ List ສອງມືຕີຂະໜາດ 3×3 ເລີ່ມຕົ້ນດ້ວຍຕຳແໜ່ງຂໍ້ມູນທີ 0 ເຊິ່ງເປັນຕຳແໜ່ງທຳອິດຕາມລວງນອນ ແລະ ຕຳແໜ່ງທຳອິດຕາມລວງຕັ້ງ ດັ່ງນີ້
`list_array[0][0]` ແລະ ຕຳແໜ່ງສຸດທ້າຍແມ່ນ `[2][2]`

```
1 list_array[0][0]
```

1

```
1 list_array[2][2]
```

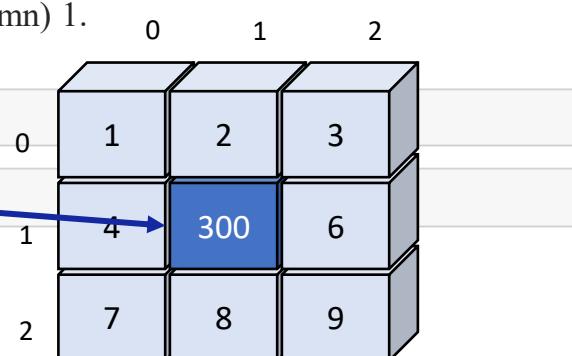
9

- ▶ 300 ແມ່ນຄ່າທີ່ໄດ້ຈາກຕຳແໜ່ງຂໍ້ມູນຕາມລວງນອນ (row) 1 ແລະ ຕຳແໜ່ງຂໍ້ມູນຕາມລວງຕັ້ງ (column) 1.

```
1 list_array[1][1] = 300
```

```
1 print(list_array[1][1])
```

300



3. ການສະແດງອົງປະກອບໃນ List ສອງມືຕີ

3.1. ການສະແດງອົງປະກອບໃນ list ສອງມືຕີ

| ການສະແດງອົງປະກອບໃນ list ສອງມືຕີຂ້ອນຂ້າງຫຍຸງຍາກ.

- ▶ ຖ້າໃຊ້ຄໍາສັ່ງກັບ list ສອງມືຕີ ດັ່ງລຸ່ມນີ້, ມັນຈະສະແດງອົງປະກອບທຳອິດຈົນເຖິງອົງປະກອບສຸດທ້າຍອອກມາ.
- ▶ ສໍາລັບກໍລະນີນີ້, ມັນຍາກທີ່ຈະເຊື້ອເຖິງຄ່າຂອງແຕ່ລະອົງປະກອບ ເຊັ່ນວ່າ 1, 2, 3.

```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
2 for item in list_array:  
3     print('item =', item)
```

```
item = [1, 2, 3]  
item = [4, 5, 6]  
item = [7, 8, 9]
```

3. ការសະແດງទីតាំងនៃការបញ្ចូលព័ត៌មានដោយប្រើប្រាស់ List

3.2. ការសະແດງការបញ្ចូលព័ត៌មានដោយប្រើប្រាស់ for loop

- | បានប្រើប្រាស់ list សម្រាប់បញ្ចូលព័ត៌មានទៅក្នុងការបញ្ចូលដោយប្រើប្រាស់ for loop.
- | ការបញ្ចូលព័ត៌មានត្រូវបានរាយការណ៍ជាអំពីការបញ្ចូលព័ត៌មានដែលត្រូវបានបញ្ជូនទៅក្នុង list.
- | list [1, 2, 3] ត្រូវបានបញ្ចូលព័ត៌មានក្នុងការបញ្ចូលដោយប្រើប្រាស់ for loop ដើម្បីបញ្ចូលព័ត៌មានទាំង 3 នេះ។
- | ប្រចាំថ្ងៃក្នុងការបញ្ចូលព័ត៌មាននេះត្រូវបានបញ្ចូលព័ត៌មានទាំង 3 នេះ។

```

1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 for i,j,k in list_array:
3     print(i,j,k)

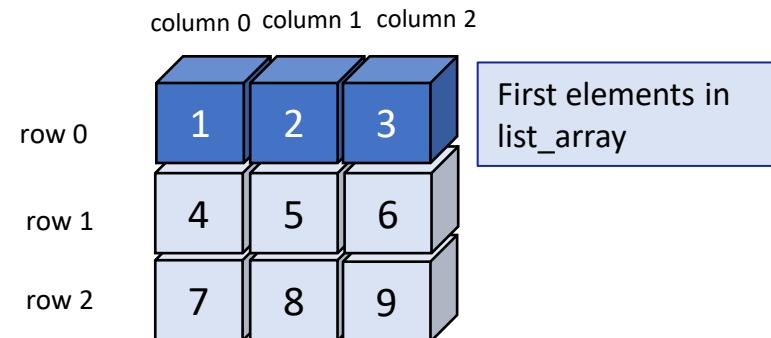
```

1 2 3
4 5 6
7 8 9

In the code below, the first elements in list_array are assigned to i, j, k

[1, 2, 3]
for i,j,k in list_array:
 print(i,j,k)

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]



3. ການສະແດງອົງປະກອບໃນ List ສອງມືຕີ

3.3. List ສອງມືຕີ ແລະ ValueError

- | ວິທີການຜ່ານມາແມ່ນມີຄວາມຖືກຕ້ອງສໍາລັບ list ທີ່ມີຄ່າແຕ່ລະອົງປະກອບເທົ່າກັນໜີດ ຄື ເທົ່າກັບສາມ.
- | ຖ້າຄ່າຂອງແຕ່ລະອົງປະກອບບໍ່ເທົ່າກັນຈະເກີດຂຶ້ນຜິດພາດຂຶ້ນ ດັ່ງສະແດງລຸ່ມນີ້.

! ValueError

```
1 list_array = [[1, 2, 3], [4, 5], [7]]  
2 for i,j,k in list_array:  
3     print(i,j,k)
```

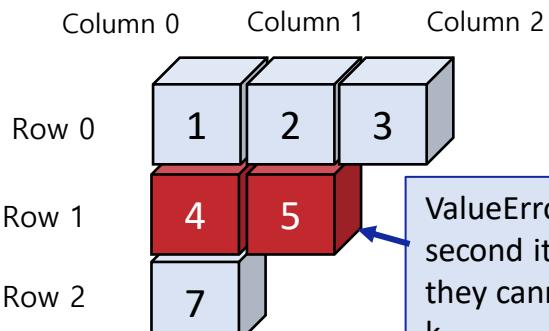
1 2 3

ValueError

Traceback (most recent call last)

```
<ipython-input-15-7a689fd851ed> in <module>  
      1 list_array = [[1, 2, 3], [4, 5], [7]]  
----> 2 for i,j,k in list_array:  
      3     print(i,j,k)
```

ValueError: not enough values to unpack (expected 3, got 2)



ValueError : Since list_array's second item has two elements, they cannot be assigned to i, j, k

3. ການສະແດງອົງປະກອບໃນ List ສອງມືຕີ

3.4. for loop ສອງຊ້ອນໃນ list ສອງມືຕີ

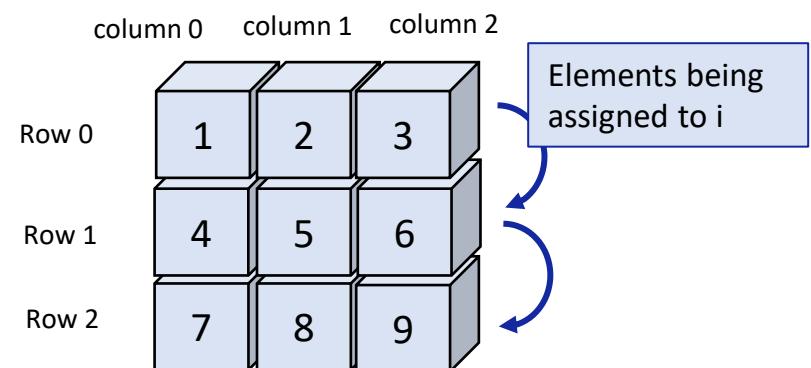
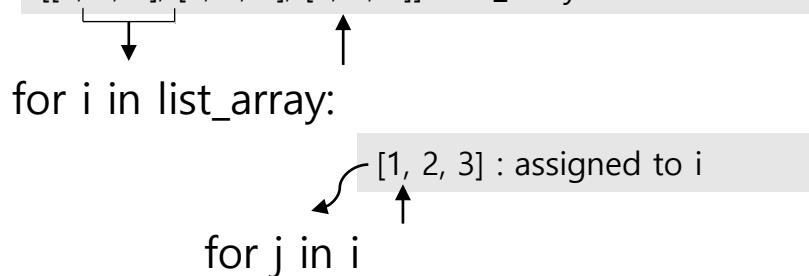
- | ທີ່ດລອງສະແດງຂໍ້ມູນໂດຍນໍາໃຊ້ for loop ສອງຊ້ອນ ດັ່ງສະແດງລຸ່ມນີ້.
- | for loop ດ້ວນນອກສໍາລັບຕົວປ່ຽນ i in list_array: ກໍານົດແຖວເທື່ອລະແຖວຈິນໜີດທຸກແຖວໃນ list.
- | ນໍາໃຊ້ for loop ດ້ວນໃນສໍາລັບຕົວປ່ຽນ j in i : ເພື່ອພິມແຕ່ລະອົງປະກອບຈາກ i.

```

1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 for i in list_array:
3     for j in i:
4         print(j, end = ' ')
5     print()

```

1 2 3
4 5 6
7 8 9



3. ការសະແດງទីតាំងនៃការបញ្ចូលព័ត៌មាននៃ List សរុប

3.5. ពិវិះយាការណ៍ដោយការបញ្ចូលព័ត៌មាននៃ List សរុប

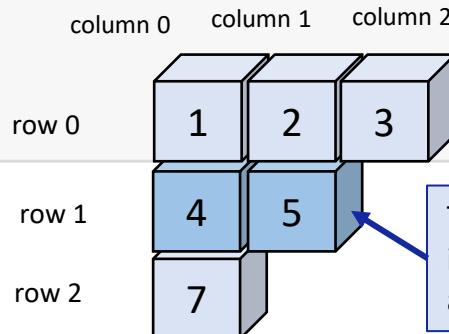
- | ទិន្នន័យនៃការបញ្ចូលព័ត៌មាននៃ List សរុប។
- | វិធីនេះ មិនធ្វើឡើងក្នុងការបញ្ចូលព័ត៌មាននៃ List សរុបទេ។

```

1 list_array = [[1, 2, 3], [4, 5], [7]]
2 for i in list_array:
3     for j in i:
4         print(j, end = ' ')
5     print()

```

1 2 3
4 5
7



There are two elements in the second item of list_array list, but the list [4, 5] is assigned to i and run without errors.

Line 2, 3, 4

- ចំណាំការបញ្ចូលព័ត៌មាននៃ list_array ដោយប្រើប្រាស់ i. ទីតាំងនៃការបញ្ចូលព័ត៌មាននៃ list_array គឺជាភាយធម៌ទី 2, 3, 4 នៅក្នុងកូដ។
- ចំណាំការបញ្ចូលព័ត៌មាននៃ i ដោយប្រើប្រាស់ j. ទីតាំងនៃការបញ្ចូលព័ត៌មាននៃ i គឺជាភាយធម៌ទី 3 នៅក្នុងកូដ។
- នៅក្នុងការបញ្ចូលព័ត៌មាននៃ i, ចំណាំការបញ្ចូលព័ត៌មាននៃ j ដោយប្រើប្រាស់ j ដែលត្រូវបានដោឡូលនៅក្នុងការបញ្ចូលព័ត៌មាននៃ i ។
- ទីតាំងនៃការបញ្ចូលព័ត៌មាននៃ j គឺជាភាយធម៌ទី 1, 2 និង 3 នៅក្នុងការបញ្ចូលព័ត៌មាននៃ i ។
- ទីតាំងនៃការបញ្ចូលព័ត៌មាននៃ i គឺជាភាយធម៌ទី 4 នៅក្នុងកូដ។

3. ການສະແດງອົງປະກອບໃນ List ສອງມືຕີ

3.6. ການພິມໂດຍນໍາໃຊ້ຕຳແໜ່ງຂໍ້ມູນ

 Focus ການພິມໂດຍນໍາໃຊ້ຕຳແໜ່ງຂໍ້ມູນ : ພິມຂະໜາດຂອງ list ສອງມືຕີ ດ້ວຍຝັງຊັ້ນ len.

- | ປ້ອນຂະໜາດຂອງແຖວ ແລະ ຖັນໃນ for in range ແລະ ພິມອົງປະກອບໃນ list.
- | len(list_array) ເທົ່າກັບ 3, ບໍ່ແມ່ນ 9.
- | ຊອກຫາຂະໜາດຂອງ list ເຊັ່ນວ່າ list_array[i] ດ້ວຍຝັງຊັ້ນ len .
- | len(list_array[i]) ແມ່ນ 3 ຕັ້ງສະແດງໃນ code ລຸ່ມນີ້.

```

1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 for i in range(len(list_array)):
3     for j in range(len(list_array[i])):
4         print(list_array[i][j], end = ' ')
5     print()

```

```

1 2 3
4 5 6
7 8 9

```

Line 2, 3

- len(list_array) : ຈະສັງແຖວທີ່ມີຂະໜາດເທົ່າ 3.
- len(list_array[i]) : ຈະສັງຂະໜາດຂອງອົງປະກອບໃນແຖວ i.

3. ການສະແດງອົງປະກອບໃນ List ສອງມືຕີ

3.7. for loop ສອງຊ້ອນໃນ list ສອງມືຕີ ດ້ວຍ item ທີ່ແຕກຕ່າງກັນ

| ວິທີການລຸ່ມນີ້ ເປັນວິທີການທີ່ໃຊ້ໄດ້ດີ ເຖິງແມ່ນວ່າຈໍານວນຂອງ item ໃນ list ຈະແຕກຕ່າງກັນ.

```
1 list_array = [[1, 2, 3], [4, 5], [7]]
2 for i in range(len(list_array)):
3     for j in range(len(list_array[i])):
4         print(list_array[i][j], end = ' ')
5     print()
```

```
1 2 3
4 5
7
```

Line 2, 3, 4

- ປະຕິບັດຊ້າສາມຄັ້ງ ໂດຍໃຊ້ range(len(list_array)).
- j ປະຕິບັດຊ້າຕາມຈໍານວນລາຍການໃນ list_array[i].
- ພິມອົງປະກອບໃນ list_array ໂດຍໃຊ້ຕຳແໜ່ງຂຶ້ນ [i][j].

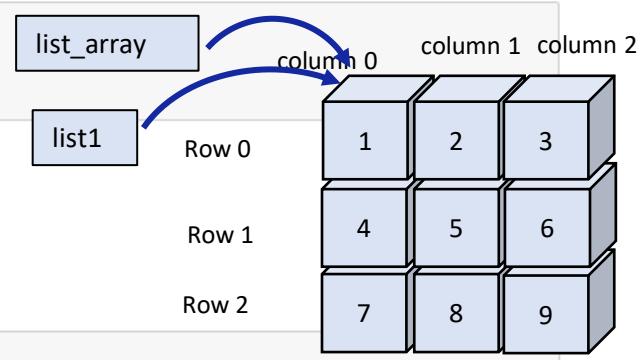
4. កំសៀវិនខ្លើនៃ list សង្គមពី

4.1. ការកំណើន list សង្គមពី និងការអ៊ាយីរី

| តូចៗថា list_array ធិនកំណើនដោយ list1 និង កំណើននៃ list1 ធិនប្រើប្រាស់ ត្រូវបានផ្តល់ទៅក្នុង list_array ដូចជាប្រើប្រាស់។

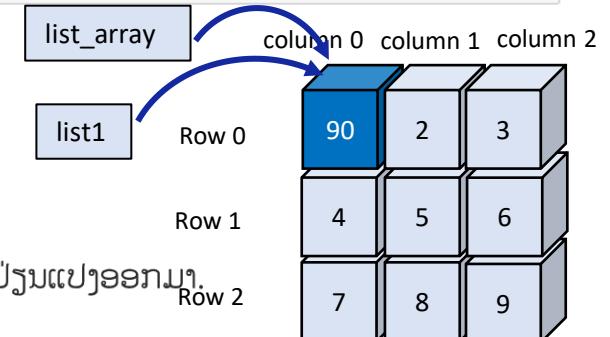
```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 list1 = list_array
3 print(list1)
```

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]



```
1 list1[0][0] = 90
2 print(list_array)
3 print(list1)
```

[[90, 2, 3], [4, 5, 6], [7, 8, 9]]
[[90, 2, 3], [4, 5, 6], [7, 8, 9]]



- ▶ ការបង្រាក់បញ្ជីបំផែនការសំណើ និង រាយការ។
- ▶ ត្រូវដើរការកំណើន list_array និង list1 ទៅក្នុង object ទីផ្សារ ដូចជាអំពីការអ៊ាយីរី
- ▶ ដើរការកំណើន list_array និង list1 ទៅក្នុង object ទីផ្សារ ដូចជាអំពីការអ៊ាយីរី



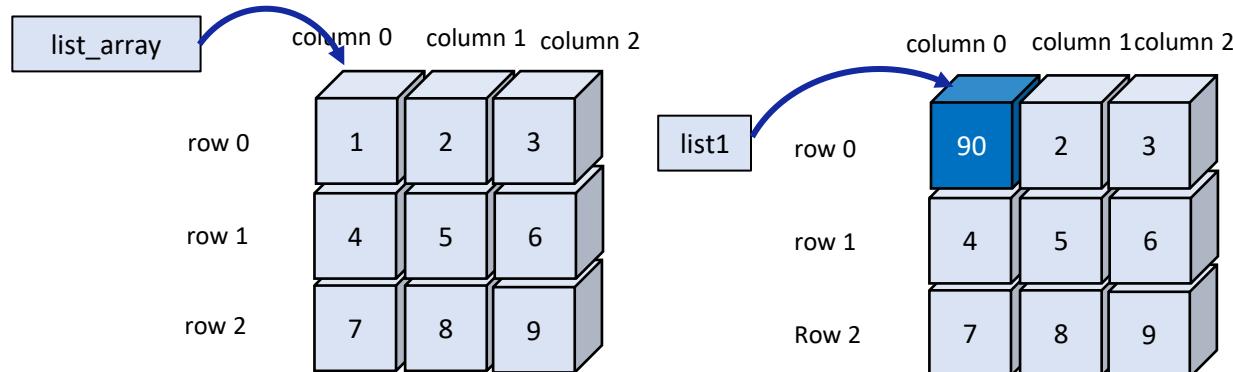
One More Step

| ถ้าต้องการสื้าๆสอบ list ในหน่วยความจำที่แตกต่างกัน จะต้องใช้ฟังก์ชัน copy.deepcopy เพื่อสำเนาไว้ใน ดั้งนี้.

```
1 import copy
2 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
3 list1 = copy.deepcopy(list_array)
4 list1[0][0] = 90
5 print(list_array)
6 print(list_array is list1)
```

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

False



Result of `copy.deepcopy` : Objects are created in separate memory.

4. ຄຳສັ່ງວິນຊື້ໃນ list ສອງມີຕີ

4.2. ນໍາໃຊ້ for loop to ເພື່ອສ້າງ list ສອງມີຕີ ສໍາລັບ 3 ແຖວ ແລະ 2 ຖັນ

```
1 list1 = []
2 for i in range(3):
3     line = []
4     for j in range(2):
5         line.append(0)
6     list1.append(line)
7
8 print(list1)
```

```
[[0, 0], [0, 0], [0, 0]]
```

Line 2, 3

- ທຳອິດ ສ້າງ list ຫວ່າງເປົ້າທີ່ຂຶ້ນ list1 ແລະ ພາຍໃນ list1 ໃຫ້ມີຂະໜາດຂອງແຖວເທົ່າກັບ 3.

Line 4, 5, 6

- ຈາກນັ້ນໃຫ້ເພີ່ມ 0 ເຂົ້າໄປ ໂດຍໃຊ້ຟັງຂັນ append, ວິນລຸບທີ່ມີຂະໜາດຂອງຖຸນເທົ່າ 2.
- ພາຍໃນລູບດ້ານນອກ, ໃຊ້ຟັງຂັນ append ອີກຄັ້ງໜຶ່ງ. ເພື່ອເພີ່ມ line ເຂົ້າໄປໃນ list1.

5. ການສ້າງ Jagged List

5.1. ຕົວຢ່າງ Jagged list

| ຈຳນວນຂອງອີງປະກອບໃນ list ສອງມີຕີ ອາດຈະແຕກຕ່າງກັນ. ສ້າງ jagged list ສອງມີຕີ ຕັ້ງສະແດງລຸ່ມນີ້.

```
1 list1 = [1, 2, 3, 4, 5]
2 list2 = []
3
4 for i in list1:
5     line = []
6     for j in range(i):
7         line.append(j)
8     list2.append(line)
9
10 print(list2)
```

```
[[0], [0, 1], [0, 1, 2], [0, 1, 2, 3], [0, 1, 2, 3, 4]]
```

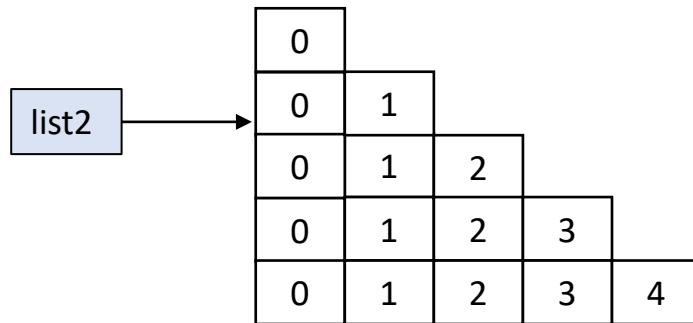
Line 4, 5, 6

- ໃນ list 1, ຂະໜາດທາງຂວາງຂອງ jagged list ຖືກກຳນົດໄວ້ກ່ອນໜ້າ.
- jagged list ຖືກສ້າງຂຶ້ນຈາກ list1. ເຊິ່ງທຳອິດຈະເປັນ list ຫວ່າງເປົ່າ.
- ປະຕິບັດຊໍ້າໃນ list 1 ໂດຍນຳໃຊ້ຟັງຊັນ for, ຮັບຄ່າແຕ່ 1 ນອດ 5 ແລະ ເກັບມັນໄວ້ໃນ i.
- ພາຍໃນ for loop, ປະຕິບັດຊໍ້າຕາມຂະໜາດລວງນອນ i ແລະ ເພີ່ມອີງປະກອບໂດຍນຳໃຊ້ຟັງຊັນ append. ຈາກນັ້ນໃຫ້ເພີ່ມອີງປະກອບ line ເຂົ້າໄປໃນ list ທີ່ຢູ່ນອກ loop.

5. ການສ້າງ Jagged List

5.1. ຕົວຢ່າງ Jagged list

| ຕົວຢ່າງລຸ່ມນີ້ແມ່ນ jagged list . jagged list ສອງມືຕີ ແລະ ຈຳນວນອີງປະກອບຂອງມັນທີ່ມີຄວາມແຕກຕ່າງກັນ.



| ສາມາດສ້າງ jagged list ທີ່ຈ່າຍ ດັ່ງສະແດງລຸ່ມນີ້.

```
1 list2 = [[0],  
2     [0, 1],  
3     [0, 1, 2],  
4     [0, 1, 2, 3],  
5     [0, 1, 2, 3, 4]]  
6 print(list2)
```

[[0], [0, 1], [0, 1, 2], [0, 1, 2, 3], [0, 1, 2, 3, 4]]



Like this, when the list size inside the list is different, it is called a jagged list.

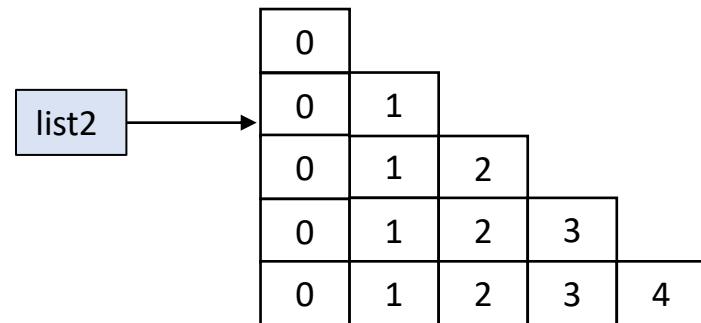
5. រាយការ Jagged List

5.2. រាយការ Jagged list តាមលំនាំខ្លួន

| រាយការ jagged list តាមលំនាំខ្លួន.

```
1 list1 = [1, 2, 3, 4, 5]
2 list2 = []
3 for i in list1:
4     line = []
5     for j in range(i):
6         line.append(j)
7         print(j, end = ' ')
8     list2.append(line)
9 print()
```

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
```



5. ການສ້າງ Jagged List

5.3. ຕົວຢ່າງການສ້າງ jagged list ໂດຍນຳໃຊ້ for loop ສອງຊ້ອນ ແລະ randint method

| ຄ່າພາຍໃນ list ສາມາດປ່ຽນເປັນຈຳນວນເຕັມແບບສຸ່ມ ດ້ວຍການນຳໃຊ້ randint(1,100).

```
1 import random
2
3 list1 = [1, 2, 3, 4, 5]
4 list2 = []
5 for i in list1:
6     line = []
7     for j in range(i):
8         line.append(random.randint(1, 100)) #takes random value
9     list2.append(line)
10
11 for i in list2:
12     for j in i:
13         print(j, end = ' ')
14     print()
```

```
79
55 96
64 17 25
97 87 7 22
73 50 9 64 71
```

Paper coding

- ຕ້ອງເຂົ້າໃຈແນວຄິດພື້ນຖານຂອງຫຼັກສູດນີ້ໃຫ້ຄົບຖ້ວນກ່ອນຈະໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.
- ການທີ່ບໍ່ເຂົ້າໃຈແນວຄິດພື້ນຖານ ຈະເພີ່ມພາລະໃນການຮຽນຮູ້ຂອງຫຼັກສູດນີ້ ແລະ ຈະເຮັດໃຫ້ບໍ່ປະສົບຜົນສໍາເລັດ.
- ມັນອາດຈະເປັນເລື່ອງທີ່ຍາກຕອນນີ້, ແຕ່ຖ້າຢາກປະສົບຜົນສໍາເລັດໄດ້ນັ້ນ ພວກເຮົາຂໍແນະນຳໃຫ້ເຂົ້າໃຈແນວຄິດພື້ນຖານ ນີ້ຢ່າງລຶກເຊິ່ງ ແລະ ກ້າວໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.

Q1.

โดยนำใช้คำเบนงชี้มุน เพื่อຊອກຫາถ่า 30 ใน list สອງມີຕີ ຄື list_array = [[10, 20], [30, 40], [50, 60]].

Conditions for Execution	30
Time	5min



Write the entire code and the expected output results in the note.

Q2.

ຈຶ່ງສ້າງ list ສອງມີຕີ ຂະໜາດ 4×4 ໂດຍໃຫ້ຄ່າຢູ່ລະຫວ່າງ 1 ເຖິງ 16 ພ້ອມທັງພິມອີງປະກອບທັງໝົດອອກມາ ໂດຍໃຊ້ for loop.

Conditions	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time	5min



Write the entire code and the expected output results in the note.

| Let's code

1. ส້າງ List ສອງມືຕີ

1.1. ส້າງ list ສອງມືຕີ ຂະໜາດ 3×2 ໂດຍນຳໃຊ້ loop statement

- | ວິນຂໍ້ແຖວທີ່ມະຂະໜາດເທົ່າ 3 ເພື່ອສ້າງ list ທົວ່າງເປົ່າ ແລະ ເພີ່ມຄ່າເຂົ້າໄປໃນ list ໂດຍນຳໃຊ້ ພັງຂັນ append.
- | ເພີ່ມ line ເຂົ້າໄປໃນ list_array ໂດຍນຳໃຊ້ append.
- | ສ້າງ list ທີ່ມີອີງປະກອບ 3 ແຖວ ແລະ 2 ຖັນ ໂດຍນຳໃຊ້ loop statement.

```
1 list_array = []      # Create empty list
2
3 for i in range(3):
4     line = []
5     for j in range(2):
6         line.append(0)
7     list_array.append(line)
8
9 print(list_array)
```

```
[[0, 0], [0, 0], [0, 0]]
```

1. ສ້າງ List ສອງມືຕີ

1.2. ສ້າງ list ສອງມືຕີ ໂດຍນຳໃຊ້ comprehension expressions

- | code ລຳສ້າງຈະຍາວຂຶ້ນອີກ ເນື່ອງຈາກໃຊ້ for loop ສອງຄັ້ງ.
- | ດ້ວຍການນຳໃຊ້ list compression expression, lis ສອງມືຕີ ສາມາດຂຽນ code ເປັນແຖວດຽວກັນໄດ້.

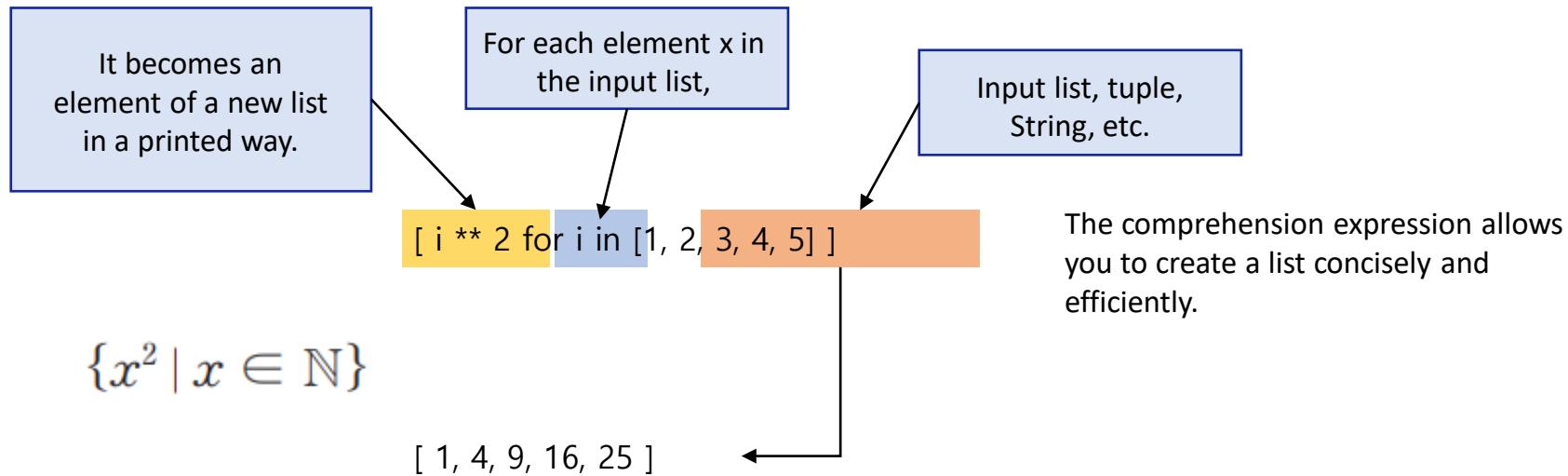
```
1 list_array = [[0] * 2 for i in range(3)]  
2 print(list_array)
```

```
[[0, 0], [0, 0], [0, 0]]
```

1. ส້າງ List ສອງມືຕີ

1.2. ส້າງ list ສອງມືຕີ ໂດຍນໍາໃຊ້ comprehension expressions

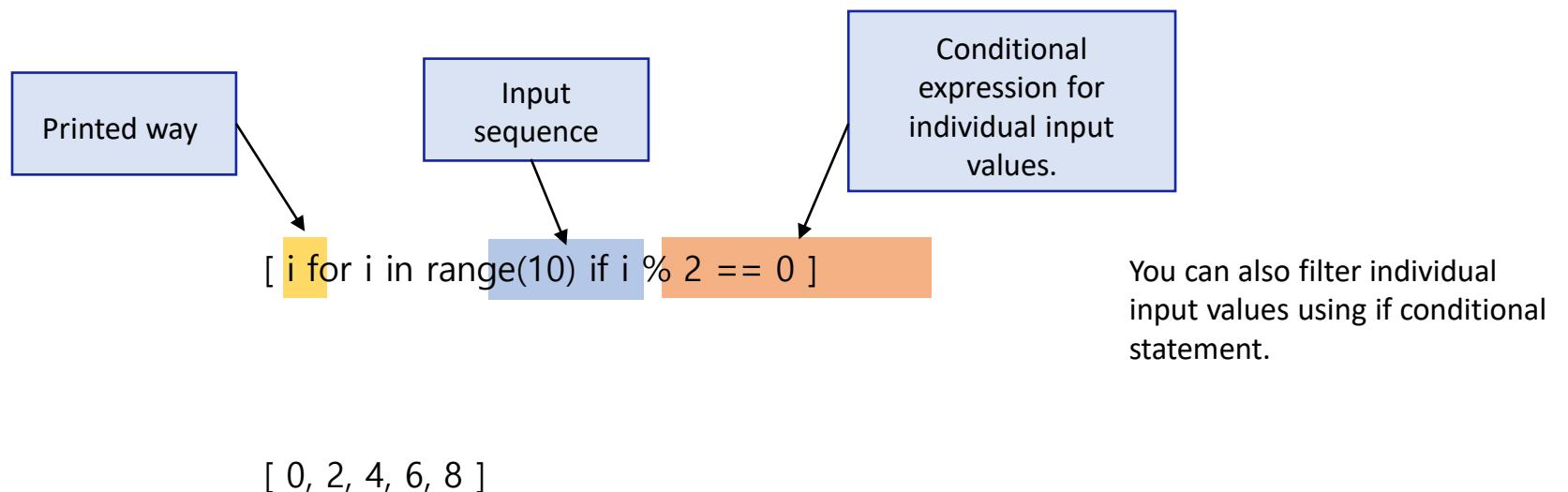
- | ໃນພາສາ Python ມີປັ້ງຂັນທີເອັນວ່າ list comprehension. ຄໍາວ່າ comprehension ໃນພາສາອັງກິດແມ່ນມີຄວາມໝາຍ implication, inclusion ແລະ intension, ມັນຄ້າຍຄືກັບນິຍາມກຸ່ມ (set) ໃນຄະນິດສາດ.
- | ການນໍາໃຊ້ list comprehension expression, ສາມາດເບິ່ງແຕ່ລະ item ທີ່ປ້ອນເຂົ້າໄປໃນ list ດັ່ງສະແດງລຸ່ມນີ້.



1. ສ້າງ List ສອງມືຕີ

1.2. ສ້າງ list ສອງມືຕີ ໂດຍນຳໃຊ້ comprehension expressions

ເຖິງອັນໄຂອາດຈະຖືກພື່ມເຂົ້າປັບໃນ list ໂດຍນຳໃຊ້ພັງຊັນ if. ເຊັ່ນຕົວຢ່າງ, ຖ້າວ່າ ກຸ່ມຂອງຕົວເລກຄູໃນລະຫວ່າງ 0 ຫາ 9 ແມ່ນ list connotation, ເຊິ່ງມັນສະແດງດັ່ງລຸ່ມນີ້.



1. ส້າງ List ສອງມືຕີ

1.3. code ຕົວຢ່າງການສ້າງ list ສອງມືຕີໂດຍນຳໃຊ້ list comprehension expression

```
1 list_array = [[0 for j in range(2)] for i in range(3)]
2 print(list_array)
[[0, 0], [0, 0], [0, 0]]
```

Line 1

- ວິນຊີ້າ 0 ສອງຄັ້ງ ນຳໃຊ້ `[0 for j in range(2)]` ເພື່ອປ່ຽນເປັນ `[0, 0]` ແລະ ວິນຊີ້າ `[0, 0]` ສາມຄັ້ງ ໂດຍນຳໃຊ້ `for i in range(3)` ເພື່ອສ້າງ `[[0, 0], [0, 0], [0, 0]]`.

2. ພິມອີງປະກອບຂອງ list ສອງມືຕີ

2.1. ພິມອີງປະກອບຂອງ list ສອງມືຕີ ໂດຍນໍາໃຊ້ while statement

- | ສາມາດໃຊ້ຝັ້ງຊັ້ນ len ເພື່ອຊອກຫາຂະໜາດຂອງ list ໂດຍ while loop statements.
- | ໃນ len(list_array), ຈະໄດ້ຮັບຂະໜາດຂອງແຖວໃນ list ສອງມືຕີ, ຈາກນັ້ນຄ່າໃນ list ທາງໃນ (ຂະໜາດຂອງຖັນ) ແມ່ນຈະຖືກກຳນົດ ແລະ ພິມອອກມາດັ່ງສະແດງລຸ່ມນີ້.

```
1 list_array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
2 i = 0  
3 while i < len(list_array):  
4     a, b, c = list_array[i]  
5     print(a, b, c)  
6  
7     i += 1
```

1 2 3
4 5 6
7 8 9

Line 2, 3

- while statement ຈະວິນຊ້າຄ່າຂອງອີງປະກອບໃນ list_array.
- ຖ້າ list_array[i] ມີອີງປະກອບ a, b, c ສະນັ້ນ i ທີ່ມີຄ່າເປັນ 0, 1, 2, 3 ຈະຖືກກຳນົດໃຫ້ອີງປະກອບ a, b, c ຕາມລຳດັບ.

3. ສ້າງ Jagged Lists

3.1. ສ້າງ ແລະ ພຶມ jagged lists

| Jagged list ຍັງສາມາດສ້າງໃຫ້ຈ່າຍກວ່າທີ່ຜ່ານມາໄດ້ອີກ ໂດຍນຳໃຊ້ comprehension expression.

```
1 list1 = [[0] * i for i in [1, 2, 3, 4, 5]]  
2 print(list1)  
  
[[0], [0, 0], [0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0, 0]]
```

| Code ລຸມນີ້ ນຳໃຊ້ for loop ສອງຊ້ອນ ເພື່ອພຶມ jagged list, list 1 ທີ່ຖືກສ້າງໃນ code ຂ້າງເທິງ ແລະ ເອີມເສັງໃນ code ດັ່ງລຸ່ມນີ້.

```
1 for i in list1:  
2     for j in i:  
3         print(j, end = ' ')  
4     print()  
  
0  
0 0  
0 0 0  
0 0 0 0  
0 0 0 0 0
```

3. ສ້າງ Jagged Lists

3.2. Append method

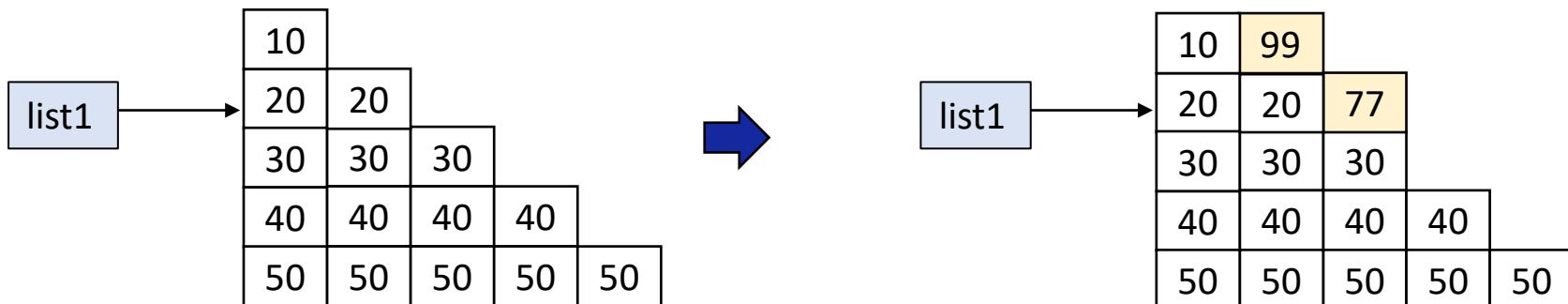
I ເພີ່ມ item ໃໝ່ ເຂົ້າໄປໃນ jagged list. ເນື່ອງຈາກ list1[0] ແລະ list1[1] ຍັງແມ່ນ lists, ອິງປະກອບ 99 ແລະ 77 ສາມາດເພີ່ມ ໂດຍນຳໃຊ້ append method.

```
1 list1 = [[10 * i] * i for i in [1, 2, 3, 4, 5]]  
2 print(list1)
```

```
[[10], [20, 20], [30, 30, 30], [40, 40, 40, 40], [50, 50, 50, 50, 50]]
```

```
1 list1[0].append(99)  
2 list1[1].append(77)  
3 print(list1)
```

```
[[10, 99], [20, 20, 77], [30, 30, 30], [40, 40, 40, 40], [50, 50, 50, 50, 50]]
```



4. อ่าน และ บันทึกข้อมูลจาก File

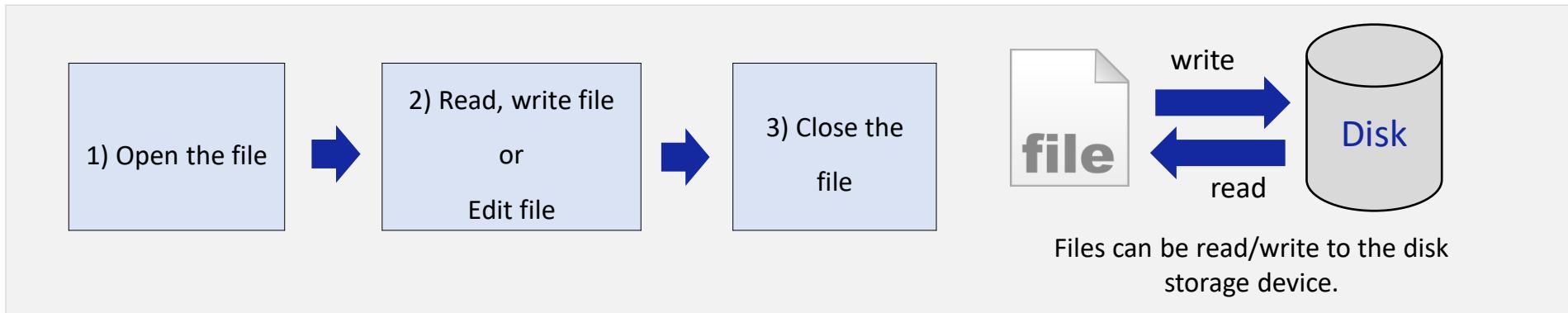
4.1. file ແມ່ນຫຍັງ?

- | ໜ່ວຍ logical ທີ່ໃຊ້ໃນການເກັບຂໍ້ມູນໃນອຸປະກອນຄອມພິວເຕີ ເຮັດວ່າ file.
- | ມັນສາມາດຈັດເກັບໄວ້ໃນອຸປະກອນຈັດເກັບຂໍ້ມູນເຊັ່ນ: hard disk ຫຼື external disk ແລະ ສາມາດເອັນໃຊ້ຂໍ້ມູນເລົ່ານັ້ນໄດ້.



4. อ่าน และ บันทึกข้อมูลจาก File

4.2. ขั้นตอนการใช้ file ใน Python



1. เปิด file

- ▶ นำ file เข้าด้วยการกำหนดที่ปุ๊ และ ต้อง file ลิงก์ในอุปกรณ์จัดเก็บข้อมูล.
- ▶ ที่ปุ๊ของ file ในอุปกรณ์จัดเก็บข้อมูลคอมพิวเตอร์ เอ็นว่า path.

2. ใช้ file ตามจุดประสงค์ที่ต้องการ

- ▶ สามารถ อ่าน, เขียน, กดสอบเมื่อใน และ เพิ่มเมื่อในเข้าไปใน file พ้อມห้างสามารถลีบเมื่อในที่มีอยู่ใน file.

3. ปิด file

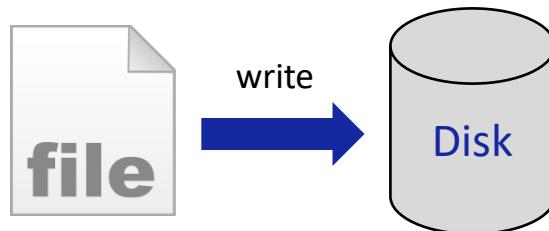
- ▶ ห้างจากเปิด file มาใช้วรากแล้ว จะต้องส่งคืนที่อยู่ความจำในระบบ.
- ▶ จะบอนการส่งคืน file ถูกกว่า เอ็นว่า ปิด file.

4. อ่าน และ บันทึกข้อมูลจาก File

4.3. กานดูรุ่น files

- | ฟังชั่น open ແມ່ນໃຊ້ເປີດ ‘hello.txt’ file.
 - ▶ ໃນນີ້ໄດ້ນຳເອົາຕົວອັກສອນ ‘w’ ໃສ່ເພື່ອເປີດ file ໃນໂທນດ ຂຽນຢ່າງດຽວ (write only)
- | ຟັງຈິນ Open ຈະສື່ງ file object.
- | File object f ນຳໃຊ້ write method ເພື່ອຂຽນ text ລົງໃນ file.
- | f.close method ເພື່ອປິດ file

```
1 f = open('hello.txt', 'w')
2 f.write('hello world!')
3 f.close()
```

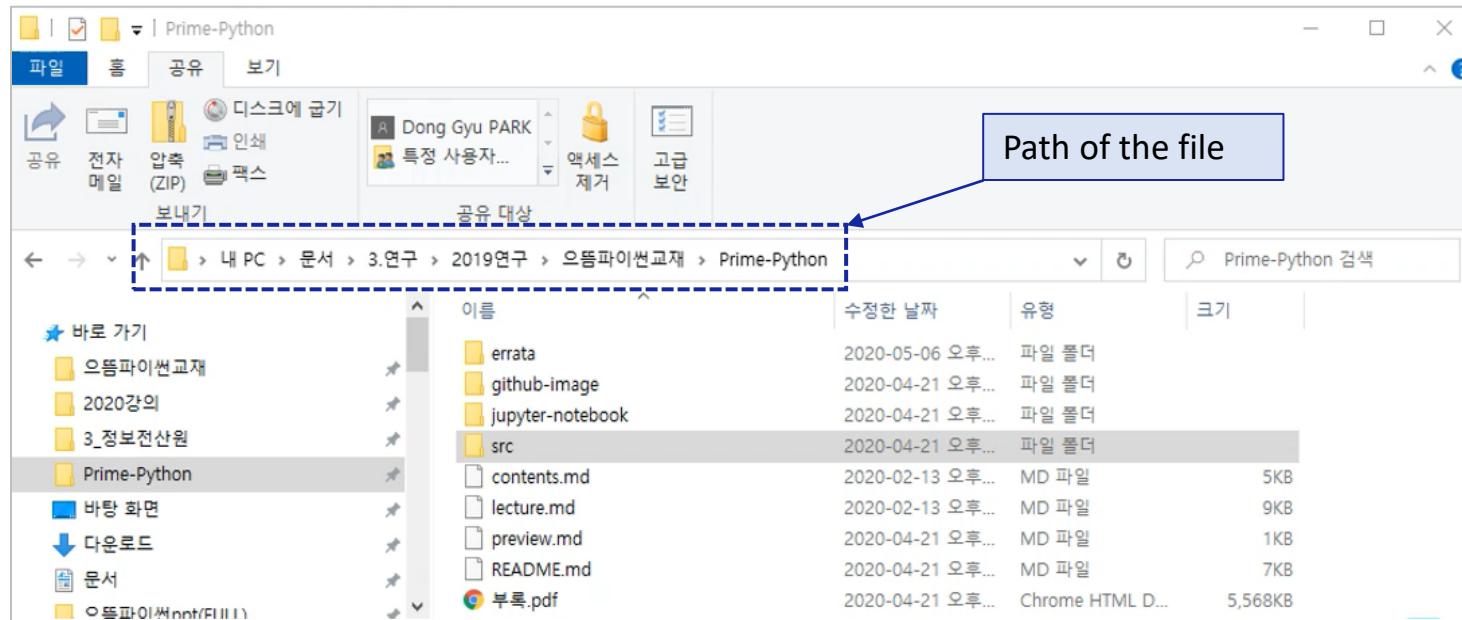


open('hello.txt', 'w') opens and brings the file in writing mode

4. อ่าน และ บันทึกข้อมูลจาก File

4.4. ฟังก์ชัน Open

- | มันเป็นฟังก์ชันที่เกี่ยวข้องกับ files ใน Python.
- | มันอาจจะมีด้วย file และ โหมดการเปิด file .
- | มันสามารถกำหนดรูปแบบของ file ผ่าน parameter เช่น mode.
- | ถ้ามีการกำหนด path ของ file, path เล่มตั้งตำแหน่งของ file แม่น Python script file.



4. อ่าน และ บันทึกข้อมูลจาก File

4.5. file input/output โหมด ในฟังก์ชัน open

Mode	Meaning
r	R หมายความจาก Read, ใช้เป็นโหมดอ่าน file ได้ป่างๆ ก្នვ (read-only mode) และ เป็นโหมดเดิมตั้งของภาษาเปิด file. - ถ้าบ่มี file มันจะเกิดຄาณ์ผิดพลาดขึ้น. - เมื่อจากมันเป็นโหมดอ่านได้ป่างๆ ก្នວ, ຄາณ์ผิดพลาดจะเกิดขึ้นเวลาเขียน file.
w	w หมายความจาก Write, ใช้เป็นโหมดเขียน file ได้ป่างๆ ก្នວ (write-only mode) - ถ้าบ่มี file มันก็จะถูกสร้างขึ้น และ ถ้ามีมันก็จะถูกเขียนทับ.
a	a หมายความจาก Append, เปิด file ในโหมดเขียน และ เพิ่มนื້ອໃນທີ່ສ້າງຂຶ້ນໃໝ່ເຂົ້າໄປ file ທີ່ມີຢູ່. - ถ้าบ่มี file ໃຫ້ສ້າງ file ໃໝ່ และ เพิ่มนີ້ອໃນ.
x	โหมดນີ້ໃຊ້ເປີດ file ສໍາລັບການຂຽນໂດຍສະເພາະ ແລະ ໃຊ້ເພື່ອສ້າງ file ໃໝ່. - ມັນບໍ່ຄືກັບໂmode a ຫຼື w, ຂີ້ຜິດພາດຈະເກີດຂຶ້ນໃນໂmodeນີ້ ເມື່ອມີ file. (Safety mode of w mode)
+	ເຄື່ອງໜາຍ + ແມ່ນ read/write ໂmode ແລະ ສາມາດໃຊ້ລວມກັນເຊັ່ນ: 'r+', 'w+', ແລະ 'r+t'. ສໍາລັບ 'r+' ສາມາດ ເປີດ ແລະ ຂຽນໃນ read ໂmode. ໃນທາງກັບກັນ 'w+' ສາມາດເປີດມັນໃນ write mode ແລະ ອ່ານມັນໄດ້.

4. อ่าน และ บันทึกข้อมูลจาก File

4.6. ໂໜນດສໍາລັບຈັດການກັບ binary files ຫຼື text files

File mode	meaning
t	t ຫຍໍ້ມາຈາກ text, ເປີດ ຫຼື ສ້າງ files ໃນຮູບແບບ text files. ໂໜນດນີ້ເປັນໂໜນດເລີ່ມຕົ້ນສໍາລັບການເປີດ file.
b	b ຫຍໍ້ມາຈາກ binary, ເປີດ ຫຼື ສ້າງ files ໃນຮູບແບບ binary.

4. ອ່ານ ແລະ ບັນທຶກຂໍ້ມູນຈາກ File

4.7. ຕົວຢ່າງ ການນຳໃຊ້ຝັງຊັນ Open

```
1 f = open('hello.txt', 'w') # Open file
```

```
1 f = open('hello.txt','wt') # Open file in text
```

```
1 f = open('hello.txt','wb') # Open file in binary
```

```
1 | f = open('hello.txt','r+t') # Text read/write mode
```

BYTE

- ‘w’ : สี่ง file object ด้วยการอ่าน file ในโหมด write (w). file object นี้ f สามารถเขียนข้อมูลใน file ที่กำหนด โดยใช้ชื่อที่ write method.
 - ‘wt’: สี่ง file object ด้วยการอ่าน file ในโหมด write (w). สามารถใช้ชื่อที่เขียนข้อมูลไปยัง file ที่กำหนด. ในนี้ สามารถเปิด และ บันทึก file ในโหมด text (t).
 - เมื่อจาก text mode เป็น default mode , มันสามารถเขียนได้.
 - ‘wb’: wb mode เป็น binary mode และ สามารถเขียน file ได้.
 - ‘r+t’: default mode เป็น read mode, แต่ยังสามารถเขียน file ได้. ใน mode นี้, ข้อผิดพลาดจะเกิดขึ้น ถ้าบ่มี file.

4. อ่าน และ บันทึกข้อมูลจาก File

4.8. ฟังก์ชัน open และ ฟังก์ชัน write

```
1 f = open('hello.txt', 'w+t') # Text write/read mode
```

```
1 f = open('hello.txt', 'a+t') # Add text
```

```
1 f.write('Hello World!') # Write in hello.txt file
```

12

```
1 f.write('Love of my life, you\'ve hurt me\n')
2 f.write('You\'ve broken my heart and now you leave me\n')
3 f.write('Love of my life, can\'t you see?\n')
```

32



Line 1

- ‘w+t’: default mode ແມ່ນ write mode, ແຕ່ຍັງສາມາດອ່ານ file ໄດ້.
- ‘a+t’: ເພີ່ມເນື້ອໃນເຂົ້າໄປແກນເນື້ອໃນຂອງ file ທີ່ມີຢູ່, ຫາຍວ່າເນື້ອໃນຫັງໝົດຂອງ file ທີ່ມີຢູ່ແມ່ນຖືກປ່ຽນແທນ.
- ມັນສາມາດອ່ານ ແລະ ສ້າງ file ຖ້າບໍ່ມີ file.

4. ອ່ານ ແລະ ບັນທຶກຂໍ້ມູນຈາກ File

4.8. ພັງຊັນ open ແລະ ພັງຊັນ write

```
1 f = open('hello.txt', 'w+t') # Text write/read mode
```

```
1 f = open('hello.txt', 'a+t') # Add text
```

```
1 f.write('Hello World!') # Write in hello.txt file
```

12

```
1 f.write('Love of my life, you\'ve hurt me\n')
2 f.write('You\'ve broken my heart and now you leave me\n')
3 f.write('Love of my life, can\'t you see?\n')
```

32



Line 1-3

- ຂຽນ text file ໂດຍນໍາໃຊ້ ພັງຊັນ f.write. ສາມາດປໍອນຂໍ້ມູນແບບ String.
- ໃນນີ້, ສາມາດຂຽນ \n ເພື່ອລົງແຖວ.
- Quotations cannot be used inside '...' indicating the beginning and end of a string. Therefore, \' can be used to print quotes.

4. อ่าน และ บันทึกข้อมูลจาก File

4.9. Close method

 **Focus** เหตุการณ์ที่_close_ method

- | โดยที่ว่าปัจจุบันในคอมพิวเตอร์จะมี buffer แล้ว ประมาณนึง ในหน่วยประมวลผลภาษา (processor) สำหรับ input/output.
- | buffer ที่ใช้ในที่นี้คือ temporary memory space.
- | ทุกครั้งที่ file ถูก upload เข้าไปที่ buffer จะมี temporary memory space, หน่วยความจำของระบบคอมพิวเตอร์จะยังคงต้องใช้ buffer ที่นี่.
- | การเขียน file ให้มีประสิทธิภาพ ต้องควบคุมข้อมูลตามขนาด และ อ่าน ที่นี่ ช่วยให้เขียนได้เร็วขึ้น.
- | ทุกเมื่อไหร่ที่ buffer ถูกส่งไป disk ด้วย close method, หน่วยความจำจะมีขนาดเท่ากับขนาดของ buffer และ ขนาดของ buffer จะหายไป.
- | ถ้าต้องการอ่าน file ที่ถูกปิดโดย close method ต้องอ่าน file ที่ถูกปิดโดย close method.

```
1 f.close() # Close file
```



One More Step

| Let's look at the need for buffers and the close method.

- ▶ Reading and writing data through a computer's hard disk is tens to hundreds of times slower than reading and writing data within a central processing unit or memory. Therefore, it is faster to collect and process reading and writing data to disk at once as much as possible. For this reason, it is faster to store 1000 bytes of data in the computer's memory 10 times in units of 100 bytes than to store 1 byte each on disk 1000 times.
- ▶ Therefore, it is generally more effective to use a buffer to collect data from the write command in units of 100 bytes and write it at once when it becomes 100 bytes. What the close method does is to store information in the buffer on disk by giving information that no more write commands are to be sent to this file.

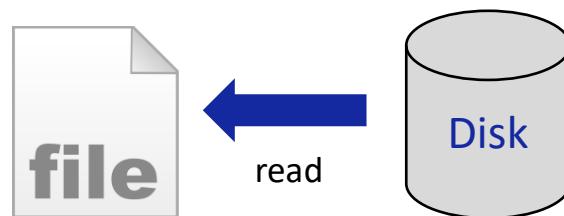
4. ອ່ານ ແລະ ບັນທຶກຂໍ້ມູນຈາກ File

4.10. ອ່ານ file

- | อ่าน file โดยนำใช้ read method.
 - | หั้งมีด strings (หั้งมีดเนื้อใน) ใน file แม่นสี่ข้อความเป็นๆ กว่าโดยบ่อมีภาระลิงๆ กว่า.

```
1 f = open('hello.txt', 'r') # Open file
2 s = f.read()             # Read hello.txt file
3 print(s)                 # print content of the file
4 f.close()                # close file
```

Hello World!Love of my life, you've hurt me
You've broken my heart and now you leave me
Love of my life, can't you see?



`open('hello.txt', 'r')` opens and brings the file in read mode.

4. อ่าน และ บันทึกข้อมูลจาก File

4.10. อ่าน file

- | ถ้าต้องการอ่าน file ให้สิ่งข้อความ 5 ตัว օอกมา โดยนำໃຊ້ f.read method ໃຫ້ປະຕິບັດດັ່ງລຸ່ມນີ້.
- | ມີພຽງ 5 ຕົວອັກສອນ ຄື 'Hello' ຖືກພິມອອກມາທາງໜ້າຈຳ.

```
1 f = open('hello.txt', 'r')      # Open file
2 s = f.read(5)                  # Read five letters in hello.txt file
3 print(s)                      # print content of the file
4 f.close()                     # close file
```

Hello



Line 1, 2

- เปิด file ใน read mode.
- ບັນເພີ່ມ 5 ແຂ້້າໄປໃນ read method, ຈະພິມຕົວອັກສອນ 5 ຕົວອອກມາ.

4. ອ່ານ ແລະ ບັນທຶກຂໍ້ມູນຈາກ File

4.11. Readline method

| อ่านข้อมูลใน file ที่ชื่อ hello.txt สອງແຖວ โดยນໍາໃຊ້ readline method ພ້ອມທັງພິມມັນອອກນາມ.

```
1 f = open('hello.txt', 'r') # Open file
2 s = f.readline()          # Read first line of the file 'AAA'
3 print(s, end = '')        # print the line
4 s = f.readline()          # Read second line of the file 'BBB'
5 print(s, end = '')        # print the line
6 f.close()                 # Close file
```

Hello World!Love of my life, you've hurt me
You've broken my heart and now you leave me

Line 2, 4

- `readline` method จะอ่านแต่หัวข้อของ file ออกมาย.
 - สามารถใช้ `readline` method ได้ทุกอย่างที่ต้องการ, เพื่ออ่านแต่ตัวไป.

4. อ่าน และ บันทึกข้อมูลจาก File

4.12. เพิ่ม file

| นำใช้ 'a+' mode

- ▶ เพิ่มเมื่อในไฟล์เดียวกับตัวท้ายเมื่อใน file ที่มีแล้ว.

| a 代表 append mode

| + 代表โหมดส้างพายໃຫຍ່ ถ้าບໍ່ມີ file.

4. ອ່ານ ແລະ ບັນທຶກຂໍ້ມູນຈາກ File

4.12. ເພີ່ມ file

```
1 f = open('hello.txt', 'a+') # Open file
2 f.write('This will be appended.\n')
3 f.write('This too.\n')
4 f.close()
```

```
1 f = open('hello.txt', 'r') # Open file
2 s = f.read()             # Read hello.txt file
3 print(s)                 # print the content of the file
4 f.close()
```

Hello World! Love of my life, you've hurt me
You've broken my heart and now you leave me
Love of my life, can't you see?
This will be appended.
This too.

} Words added



Line 2, 3

- Add line-breaking character (\n) and strings using write method.

4. อ่าน และ บันทึกข้อมูลจาก File

4.13. ติ่อย่างงานอ่าน และ เขียน file

- | มาຊອກຫາ applications.
- | ຮັບຄ່າທີ່ເປັນຈຳນວນຕຸວນ 5 ຄ່າ ແລະ ເກັບພວກມັນລົງໃນ data5.txt.
- | อ่าน data5.txt ແລະ ພິມຜົນບວກ ພ້ອມຄ່າສະເລ່ຍຂອງຄ່າດັ່ງກ່າວ.

```
1 f = open('data5.txt', 'w')                      # Open file in writing mode
2 for _ in range(5):
3     n = input('Enter integer: ')
4     f.write(n)                                     # Write the input in string
5     f.write('\n')                                 # Write the value and change line
6 f.close()                                         # close file
```

Enter integer: 10
Enter integer: 20
Enter integer: 30
Enter integer: 40
Enter integer: 50

4. อ่าน และ บันทึกข้อมูลจาก File

4.13. ติวอย่างงานอ่าน และ เขียน file

| เวลาป้อนค่า 10, 20, 30, 40, 50 แล้วให้รีปกวดเบื้อง Jupiter homepage กำจะเห็น file ที่ชื่อ data5.txt file ถูกสร้างขึ้น ตั้งสะແດງลุ่มນี้.

<input type="checkbox"/>	<input type="checkbox"/> Favorites	2년 전	
<input type="checkbox"/>	<input type="checkbox"/> jupyter-work	18일 전	
<input type="checkbox"/>	<input type="checkbox"/> Links	9달 전	
<input type="checkbox"/>	<input type="checkbox"/> Music	9달 전	
<input type="checkbox"/>	<input type="checkbox"/> OneDrive - 창원대학교	12일 전	
<input type="checkbox"/>	<input type="checkbox"/> PycharmProjects	3달 전	
<input type="checkbox"/>	<input type="checkbox"/> Saved Games	한 달 전	
<input type="checkbox"/>	<input type="checkbox"/> scikit_learn_data	3달 전	
<input type="checkbox"/>	<input type="checkbox"/> Searches	9달 전	
<input type="checkbox"/>	<input type="checkbox"/> source	2년 전	
<input type="checkbox"/>	<input type="checkbox"/> Videos	9달 전	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Chapter1.ipynb	Running 9시간 전	126 kB
<input type="checkbox"/>	<input checked="" type="checkbox"/> Chapter2.ipynb	Running 4분 전	231 kB
<input type="checkbox"/>	<input checked="" type="checkbox"/> Untitled.ipynb	18일 전	43.3 kB
<input type="checkbox"/>	<input checked="" type="checkbox"/> Untitled1.ipynb	18일 전	1.07 kB
<input type="checkbox"/>	<input type="checkbox"/> 1.10	5달 전	155 B
<input type="checkbox"/>	<input type="checkbox"/> 1.18.5	5달 전	155 B
<input checked="" type="checkbox"/>	<input type="checkbox"/> data5.txt	6분 전	20 B
<input checked="" type="checkbox"/>	<input type="checkbox"/> hello.txt	12분 전	228 B

4. อ่าน และ บันทึกข้อมูลจาก File

4.13. ตัวอย่างงานอ่าน และ เขียน file

- | อ่านตัวเลขจาก file และ พิมพ์ในบวก ผู้อ่านถ้าจะเลือกเลื่อนหน้า.
- | เมื่อจาก 10 ตัวบันจุ่ย์ใน file ที่เป็นรูปแบบ string "10", งานคำนวณงาน งานเพิ่ม จะเพิ่มได้ต้องใช้ฟังก์ชัน int.

```

1 f = open('data5.txt', 'r')                      # open file in reading mode
2 su = 0
3 for _ in range(5):
4     n = int(f.readline())                      # read a number within a line of data5.txt file
5     su += n                                     # accumulate and add to su.
6
7 print('Sum of the numbers = {}, average = {}'.format(su, su/5))    # print sum and average
8 f.close()                                       # close file

```

Sum of the numbers = 150, average = 30.0

Line 4, 5

- n เป็นจำนวนโดยใช้ readline ที่เป็นประเภท int.
- นำมิตา n ให้加บ su และบวก su แล้ว加บ su ทั้ง su+n เข้าไปต่อ su คือว่า.

4. อ่าน และ บันทึกข้อมูลจาก File

4.14. With statement

| with ~ as : สามารถมำใช้เพื่อ呼្មານ และ ปิด file ได้ย່າງຍ່າຍດາຍ.

```
1 with open('hello.txt', 'w') as f:           # automatically open and close file
2     f.write('Hello World!')                   # write in hello.txt file
```

| ບໍ່ຈໍາເປັນຕ້ອງໃສ່ສ່ວນສຸດທ້າຍ ເນື່ອງຈາກ f.close จะດຳເນີນການໂດຍອັດຕະໂນມັດຫຼັງຈາກຂຽນ file ສໍາເລັດ.



One More Step

I With statement in Python

- ▶ The with statement is a way to clarify grammar and handle exceptions easily.
- ▶ It defines the `__enter__` and `__exit__` functions executed by the context manager, replacing the code executed at the front and back of the with syntax body.
- ▶ Using the with statement, concise code is possible on behalf of try-except-finally.
- ▶ Since the `f.write` operation always contains the possibility of errors, it must be coded as follows using the try clause.
- ▶ It is also important to close the file, so you must put this code after the final clause.
- ▶ Such cumbersome work can be done simply using with statement.

```
1 f = open('hello.txt', 'w')                      # open file
2 try:                                            # write in hello.txt file
3     f.write('Hello World!')
4 finally:
5     f.close()                                     # close file
```



One More Step

I What is context manager?

- ▶ The context manager is a Python class that plays a role in accurately allocating and providing resources at the desired timing.
- ▶ Computer works using memory and CPU, and these computing elements are called resources.
- ▶ If this resource is not properly managed, the computer will not work.
- ▶ The main class that manages these resources is Python's context manager.
- ▶ Closing after opening a file is the main role of the resource management program, so the context manager plays this role in Python.
- ▶ The try-except-finally on the previous page is described in Unit 19 as an exception handling statement.
- ▶ Details on these are beyond the scope of this course.



One More Step

- | With statement is also useful for simplifying the syntax that accesses the page by opening a web page as follows.
- | To open a web page, use a package called urlopen. You can import the contents of a web page through this package. In this case, the with statement is useful as well.
- | The following code is a code that brings the content of a web page called python.org.

```
1 import urllib.request  
2  
3 with urllib.request.urlopen('http://python.org/') as response:  
4     html = response.read()  
5     print(html)
```

Python Context Manager returns `HTTPResponse` object and calls a `close` method that automatically terminates Internet access after this section.
(Otherwise, this Python program continues to access the Internet and wastes memory.)

```
b'<!doctype html>\n<!--[if lt IE 7]> <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9"> <![endif]-->\n<!--[if IE 7]>  
<html class="no-js ie7 lt-ie8 lt-ie9"> <![endif]-->\n<!--[if IE 8]> <html class="no-js ie8 lt-ie9">  
<![endif]-->\n<!--[if gt IE 8]><!--><html class="no-js" lang="en" dir="ltr"> <!--<![endif]-->\n\n<head>\n    <meta char  
set="utf-8">\n    <meta http-equiv="X-UA-Compatible" content="IE=edge">\n    <link rel="prefetch" href="//ajax.googlea  
pis.com/ajax/libs/jquery/1.8.2/jquery.min.js">\n    <link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/jqueryui/  
1.12.1/jquery-ui.min.js">\n    <meta name="application-name" content="Python.org">\n    <meta name="msapplication-tool  
tip" content="The official home of the Python Programming Language">\n    <meta name="apple-mobile-web-app-title" conten  
t="Python.org">\n    <meta name="apple-mobile-web-app-capable" content="yes">\n    <meta name="apple-mobile-web-app-stat  
us-bar-style" content="black">\n    <meta name="viewport" content="width=device-width, initial-scale=1.0">\n    <meta
```

| Pair programming



Pair Programming Practice

| ແນວທາງ, ກົມໄກ ແລະ ແຜນສຸກເສີນ

ການຈັບຄຸ້ຂຽນໂປຣແກຣມ ເປັນການຈັບຄຸ້ຂອງນັກຮຽນເພື່ອຮັດວຽກມອບໝາຍ, ນັກຮຽນຄວນມີແຜນ ແລະ ສາມາດປ່ຽນແທນກັນໄດ້ ໃນ ກໍາລະນີມີຜູ້ໃຫ້ນີ້ບໍ່ສາມາດເຂົ້າຮ່ວມຮັດວຽກມອບໝາຍໄດ້ບໍ່ວ່າໃນກໍາລະນີໃດກຳຕາມ ເຊິ່ງບັນຫາເລື່ອນັ້ນຕ້ອງຮັດໃຫ້ຈະເຈັ້ງ ແລະ ກຳບໍ່ແມ່ນ ຄວາມຜິດຂອງນັກຮຽນທີ່ຈັບຄຸ້ບໍ່ດີ.

| ຈັບຄຸ້ທີ່ຄ້າຍຄືກັນ, ບໍ່ຈໍາເປັນເຫົ້າທຽມກັນ, ຄວາມສາມາດເປັນຄຸ້ຮ່ວມງານ

ການຈັບຄຸ້ຂຽນໂປຣແກຣມ ຈະໄດ້ຮັບຜົນທີ່ກຳຕໍ່ເນື້ອນັກຮຽນມີຄວາມສາມາດຄ້າຍຄືກັນ ຫາຍວ່າ ບໍ່ຈໍາເປັນຕ້ອງມີຄວາມສາມາດຄືກັນກຳໄດ້, ແຕ່ວ່າ ການຈັບຄຸ້ ນັກຮຽນທີ່ມີຄວາມສາມາດແຕກຕ່າງກັນຫຼາຍ ກໍຈະຮັດໃຫ້ບໍ່ສົມດຸນກັນ. ຄຸສອນຮູ້ດີວ່າ ການຈັບຄຸ້ກັນບໍ່ແມ່ນ ຍຸດທະສາດ “ແບ່ງເພື່ອເອົາຊະນະ” ແຕ່ເປັນຄວາມ ພະຍາຍາມຮັດວຽກຮ່ວມກັນຂອງນັກຮຽນໃຫ້ປະສິບຜົນສໍາເລັດ. ຄຄວນທີ່ກາເວັ້ນການຈັບຄຸ້ກັນລະຫວ່າງນັກຮຽນອ່ອນ ແລະ ນັກຮຽນເກົ່າງ.

| ກະຕຸ້ນນັກຮຽນໂດຍການໃຫ້ສິ່ງຈຸງໃຈພື້ນເສດ

ຂໍສະເໜີແຮງຈຸງໃຈທີ່ຮັດໃຫ້ນັກຮຽນຈັບຄຸ້, ໂດຍສະເພາະນັກຮຽນທີ່ມີຄວາມສາມາດສຸງ. ບາງຄຸສອນໄດ້ພື້ນວ່າ ການຈັບຄຸ້ຮັດວຽກມອບໝາຍ ແມ່ນມີ ປະໂຫຍດ ສໍາລັບໜຶ່ງ ຫຼື ສອງວຽກມອບເທົ່ານັ້ນ



Pair Programming Practice

| ป้องกันงานบ่ตั้งในระบบของมัคฐาน

สิ่งท้าทายสำคัญແມ່ນເພື່ອຊອກຫາວິທີທີ່ຈະປະເມີນຜົນການຮຽນຂອງມັກຮຽນ, ຄຸນບໍ່ວ່າ ມັກຮຽນ ໄດ້ຕັ້ງໃຈຮຽນ ຫຼື ບໍ່ຕັ້ງໃຈຮຽນ. ຜູ້ສ່ວວຊານໄດ້ແນະນຳໃຫ້ທີບທວນການອອກແບບຫຼັກສຸດການຮຽນ ແລະ ຮູບແບບການປະເມີນ ພ້ອມທັງປີກາຫາລືຢ່າງຈົງຈ້າງກັບມັກຮຽນ ກ່ຽວກັບພິດຕິກຳທີ່ຈະບໍ່ຕັ້ງໃຈຮຽນ ນອກຈາກນີ້ຍັງໄດ້ແນະນຳມອບວຽກມອບໝາຍໃຫ້ມັກຮຽນ ພ້ອມທັງອະທິບາຍໃຫ້ເຂົ້າເຈົ້າຢ່າງຈະເຈັ້ງ

| ສະພາບແວດລ້ອມຂອງການຮຽນຮູ້ຮ່ວມກັນ

ສະພາບແວດລ້ອມການຮຽນຮູ້ຮ່ວມກັນເກີດຂຶ້ນໄດ້ທຸກເວລາທີ່ຜູ້ສອນຮຽກຮ້ອງໃຫ້ມັກຮຽນເຮັດວຽກຮ່ວມກັນໃນກິດຈະກຳການຮຽນຮູ້ ເຊິ່ງ ອາດຈະເປັນກິດຈະກຳທີ່ເປັນທາງການ ແລະ ບໍ່ເປັນທາງການ ແລະ ອາດຈະບໍ່ລວມເຖິງການປະເມີນຜົນການຮຽນໂດຍກິງ. ເຊັ່ນຕົວຢ່າງ ໃຫ້ ມັກຮຽນຈັບຄຸ້ມັກເພື່ອເຮັດວຽກມອບໝາຍ ໂດຍມັກຮຽນຈະຕ້ອງທີບທວນກ່ຽວກັບການສອນຂອງອາຈານທີ່ຜ່ານມາ ແລະ ລະດົມແນວຄິດພາຍໃນກຸ່ມ ພ້ອມທັງມືການແບ່ງວຽກໃຫ້ແຕ່ລະຄົນຮັບຜິດຊອບ ຈາກນັ້ນກໍໃຫ້ມີການແລກປ່ຽນຄວາມຄິດເຫັນເຊິ່ງກັນ ແລະ ກັນ ເພື່ອເຮັດວຽກມອບໝາຍໃຫ້ສໍາເລັດຕາມເປົ້າໝາຍທີ່ວ່າງໄວ້.

Q1.

Write a program that generates a multidimensional array of $n \times n$ size, based on the number of inputs, by receiving two or more n as inputs from users. In this case, the content of the arrangement should be displayed so that the values of 0 and 1 intersect in a checkered pattern.

Output example

Enter n: 5

```
1 0 1 0 1  
0 1 0 1 0  
1 0 1 0 1  
0 1 0 1 0  
1 0 1 0 1
```