Unit 7.

ຄຳສັ່ງແບບມີເງື່ອນໄຂ 2: ການຕັດສິນໃຈໃນສອງທິດ ທາງ ແລະ ນຳໃຊ້ສຳນວນທີ່ມີເງື່ອນໄຂ

Learning objectives

- √ ເຂົ້າໃຈຄຳສັ່ງແບບມີເງື່ອນໄຂຕ່າງໆເຊັ່ນ: if statement, if-else statement, ແລະ ifelse-elif statement ແລະ code ຕາມສະຖານະການ.
- √ ແກ້ໄຂບັນຫາທີ່ຊັບຊ້ອນຫຼາຍ, ໂດຍນຳໃຊ້ຄຳສັ່ງ double if-else.
- √ ການແກ້ໄຂ ແລະ ການປະຍຸກໃຊ້ວິທີແກ້ບັນຫາຕ່າງໆໂດຍຜ່ານຄຳສັ່ງ if-else ແລະ logic, ຕົວດຳເນີນການປຸງບທຸງບ.
- 🗸 ສຸ່ມຕົວເລກຕາມໃຈ ໂດຍໃຊ້ random modules.

Lesson overview

- √ ໃຊ້ສຳນວນທີ່ເໝາະສົມສຳລັບຄຳສັ່ງ if, elif.
- √ ກວດສອບສຳນວນເງື່ອນໄຂຂອງ elif ເມື່ອສຳນວນເງື່ອນໄຂຂອງຄຳສັ່ງ if ບໍ່ເປັນ ຈິງ.
- √ ທຳຄວາມເຂົ້າໃຈຄຳສັ່ງການດຳເນີນການຂອງ if-elif-else ແລະ ນຳໃຊ້ເພື່ອແກ້ໄຂບັນຫາຕ່າງໆ.
- √ ທຳຄວາມເຂົ້າໃຈວ່າ ສຳນວນເງື່ອນໄຂມີຄ່າໃນປະເພດ Boolean ເປັນ True ແລະ False, ພ້ອມທັງແກ້ໄຂບັນຫາດ້ວຍຕົວດຳເນີນການຕ່າງໆ.

Concepts You Will Need to Know From Previous Units

- 🗸 ທຳຄວາມເຂົ້າໃຈຄຳສັ່ງຄວບຄຸມການດຳເນີນການຕາມເງື່ອນໄຂອື່ນ ນອກເໜືອຈາກການດຳເນີນການຕາມລຳດັບ.
- 🗸 ທຳຄວາມເຂົ້າໃຈວ່າຄຳສັ່ງເງື່ອນໄຂ ເປັນຄຳສັ່ງທີ່ດຳເນີນການເມື່ອກົງກັບເງື່ອນໄຂທີ່ກຳນົດໄວ້ເທົ່ານັ້ນ.
- √ ທຳຄວາມເຂົ້າໃຈຂັ້ນຕອນດຳເນີນການ ຂອງຕົວດຳເນີນການແບບ Boolean.

Keywords

if - else

if-else-elif

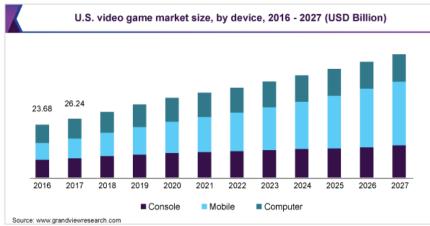
Double Conditional
Statement

Random
Numbers

Mission

1. Real world problem

1.1 ຂະໜາດຂອງຕະຫຼາດເກມທົ່ວໂລກ



https://www.grandviewresearch.com/industry-analysis/video-game-market

ນອກຈາກນີ້ເມື່ອມີຫຼາຍຄົນມາມ່ວນຊື່ນກັບ ການຫຼິ້ນເກມໃນເຮືອນ ເນື່ອງຈາກການກັກ ຕົວຍ້ອນ COVID-19, ຄວາມຮັບຮູ້ທາງລົບ ຕໍ່ເກມກໍດີຂຶ້ນຫຼາຍ.

- ໃນປີ 2020, ຂະໜາດຕະຫຼາດຂອງເກມທົ່ວໂລກແມ່ນ
 \$186.491 ຕື້, ເພີ່ມຂຶ້ນ 5.0% ຈາກປີທີ່ຜ່ານມາ.
- ອຸດສາຫະກຳເກມເຫຼົ່ານີ້ກຳລັງຈະກາຍເປັນອຸດສາຫະກຳ ຂະໜາດໃຫຍ່ໃນບັນດາປະເທດຕ່າງໆເຊັ່ນ: ອາເມລິກາ,
 ຈີນ, EU, ສ.ເກົາຫຼີ, ຍີ່ປຸ່ນ.
- ຮູບເບື້ອງຊ້າຍສະແດງໃຫ້ເຫັນຂະໜາດຕະຫຼາດຂອງວິດີ ໂອເກມໃນສະຫະລັດ. ຈາກຂໍ້ມູນຂອງສະຖາບັນການວິໄຈ, ຕະຫຼາດວິດີໂອເກມຄາດວ່າຈະມີການຂະຫຍາຍຕົວຢ່າງຕໍ່ ເນື່ອງໃນແຕ່ລະປີ.

1. Real world problem

1.2. FIFA Online 4



- EA Sports ເປັນຜູ້ພັດທະນາເກມລະດັບໂລກ ແລະ ມີ ຊື່ສຸງຫຼາຍສໍາລັບ FIFA series.
- ເກມບານເຕະອອນລາຍ FIFA Online 4, ເປີດຕົວ ໂດຍ EA Sports ໃນປີ 2018, ໄດ້ຮັບການຕອບຮັບ ເປັນຢ່າງດີສໍາລັບຮູບພາບກຣາຟິກ ແລະ ການເຄື່ອນ ໄຫວທີ່ສົມຈິງ. ຮຽນແບບການເຕະບານຕົວຈິງ, ຄວາມ ເພື່ງພໍໃຈຂອງຜູ້ຊົມ, ການດໍາເນີນການຂອງສະໜາມ ແລະ ອື່ນໆ.
- ໃນກິດຈະກຳນີ້, ພວກເຮົາຈະສ້າງຟັງຊັນການຍິງຈຸດ
 ໂທດຂອງກິລາບານເຕະ, ໂດຍໃຊ້ຟັງຊັນຂັ້ນສູງຂອງຄຳ ສັ່ງແບບມີເງື່ອນໄຂ.

1. Real world problem

1.3. Penalty shootout rules



ຄອມພິວເຕີຈະເປັນຜູ້ຮັກສາປະຕູ
 ແລະ ປ້ອງກັນ, ໂດຍການສ້າງຕົວ
 ເລກແບບສຸ່ມເພື່ອປ້ອງກັນໜຶ່ງໃນ
 ສາມພື້ນທີ່

- ການຍິງຈຸດໂທດແມ່ນກົດລະບຽບທີ່ໃຊ້ໃນການແຂ່ງຂັນ ບານເຕະ ເມື່ອຄະແນນສະເໝືກັນ, ເຖິງແມ່ນວ່າຈະຕໍ່ ເວລາກໍຕາມ, ໃນບາງກໍລະນີເອີ້ນວ່າ PK (Penalty Kick), ເຊິ່ງເປັນຕົວຢ່າງຂອງການເຕະລູກໂທດ, ແຕ່ ຄຳທີ່ຖືກຕ້ອງຄືການຍິງຈຸດໂທດ. ດັ່ງນັ້ນຈຶ່ງຖືກຕ້ອງທີ່ ຈະເຮັດເຄື່ອງໝາຍເປັນ PS ຫຼື PSO.
- ພວກເຮົາຈະຈຳລອງເກມດັ່ງກ່າວໂດຍໃຊ້ຄອມພິວເຕີ.
- ສົມມຸດວ່າຜູ້ໃຊ້ພະຍາຍາມເຕະລູກໂທດກັບຄອມພິວ ເຕີ. ໃນກໍລະນີນີ້ຖືວ່າຜູ້ໃຊ້ອາດຈະເລືອກໃຊ້ພື້ນທີ່ໃດ ໜຶ່ງຈາກ 3 ພື້ນທີ່ໄດ້ແກ່: ຊ້າຍ, ກາງ, ຂວາ ແລະ ທຳການເຕະລູກໂທດ.

- 1. Real world problem
- 1.4. ຂໍ້ມູນອ້າງອີງເພີ່ມເຕີມສໍາລັບການຍິງຈຸດໂທດ

l ກວດສອບເບິ່ງ Wikipedia, https://en.wikipedia.org/wiki/Penalty_shootout_(association_football) ສໍາລັບຂໍ້ມູນເພີ່ມເຕີມ.



https://en.wikipedia.org/wiki/Penalty_shoot-out_(association_football)

2. Mission

2.1. ຂັ້ນຕອນຂອງເກມຍິງຈຸດໂທດ

```
import random
3 | n = random.randint(1, 3) # random7y generates numbers among 1,2,3
4 if n == 1:
       computer choice = "Left"
  elif n == 2:
       computer choice = "Middle"
8 else:
       computer choice = "Right"
9
10
11 | user_choice = input("which side will you attack?(left, middle, right) : ")
12 | if computer_choice == user_choice:
       print("offense failed.")
14 else :
       print("Congrats!! Offense succeeded.")
16 print('computer defense position:', computer_choice)
```

```
which side will you attack? (left, middle, right): right
Congrats!! Offense succeeded
'computer defense position : left
```

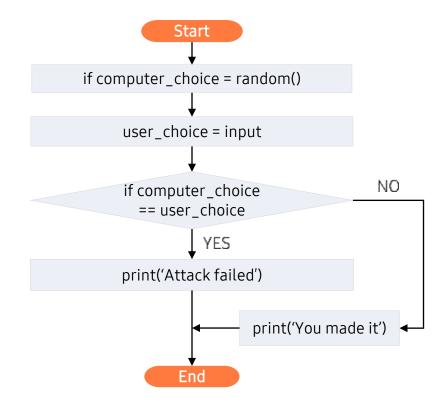
2. Mission

2.2. ແຜນການຂຸງນໂປຣແກຣມ (Programming plan)

Pseudocode

- [1] Start
- [2] Receive random input.
- [3] User enters one of the three inputs.
- [4] Check if the direction of the keeper and the direction of the ball are identical.
- [5] if identical, print 'Attack failed'
- [6] if different, print 'Offense succeeded.'
- [7] End

Flowchart



2. Mission

2.3. code ສຸດທ້າຍຂອງເກມຍິງຈຸດໂທດ

```
1 import random
n = random.randint(1, 3) # randomly generates numbers among 1,2,3
4 if n == 1:
      computer_choice = "Left"
6 elif n == 2:
      computer_choice = "Middle"
   else:
      computer_choice = "Right"
10
11 | user_choice = input("Which side will you attack?(left, middle, right) : ")
12 | if computer_choice == user_choice:
      print("Offense failed.")
14 else :
      print("Congrats!! Offense succeeded.")
16 | print('computer defense position :', computer_choice)
```

Key concept

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.1. ຄຳສັ່ງຂອງ else ຈະຖືກປະຕິບັດເມື່ອເງື່ອນໄຂບໍ່ເປັນຈິງ
- l Python ໄດ້ກະກຸງມຄຳສັງ if-else ສຳລັບໂຄງສ້າງການຄັດເລືອກ. ສົມມຸດວ່າໄດ້ຮັບການຕອບ ຮັບ ແລະ ໄດ້ຮັບທຶນການສຶກສາ, ຖ້າວ່າໄດ້ຄະແນນຫຼາຍກວ່າ ຫຼື ເທົ່າກັບ 90. ທາງກົງກັນຂ້າມ, ຈະບໍ່ໄດ້ຮັບທຶນການສຶກສາຖ້າຄະແນນຕໍ່າກວ່າ 90.
 - Code as follows in Python.

```
1 if score > 90:
     print('Accepted')
     print('Also receive a scholarship')
     print('Not accepted.')
```

- ປະໂຫຍກ if-else ແມ່ນປະໂຫຍກທີ່ວ່າ "ຖ້າເງື່ອນໄຂເປັນຈິງ, ໃຫ້ປະຕິບັດຄຳສັ່ງນີ້ ແລະ ຖ້າ ເງື່ອນໄຂບໍ່ເປັນຈິງໃຫ້ປະຕິບັດຕາມຄຳສັ່ງ else.
- l ບົດຮຸງນນີ້ຈະຈັດການກັບຄຳສັ່ງທີ່ມີເງື່ອນໄຂທີ່ສັບຊ້ອນຫຼາຍໃນລັກສະນະນີ້

Key concept

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.2. ຄຳສັ່ງ if ໂດຍທີ່ບໍ່ມີຄຳສັ່າອື່ນ
- l ນີ້ແມ່ນສະຖານະການໃໝ່. ພິຈາລະນາໂປຣແກຣມທີ່ກຸ່ງວຂ້ອງກັບສະຖານະການຂ້າງລຸ່ມນີ້ ທີ່ຖືກດຳ ເນີນການໂດຍຄຳສັ່ງ if ຈາກບົດຮຸເນທີ່ຜ່ານມາ

ຟັງຊັນພິມ "It is morning" ກ່ອນ 12 pm, ແລະ "It is afternoon' ຫຼັງຈາກ 12 pm

l ຖ້າໃຊ້ພຸງແຕ່ຄຳສັ່ງ if, ກໍຈະສາມາດສ້າງ Code ໄດ້, ໂດຍໃຊ້ ifs ສອງຕົວດັ່ງຕໍ່ໄປນີ້

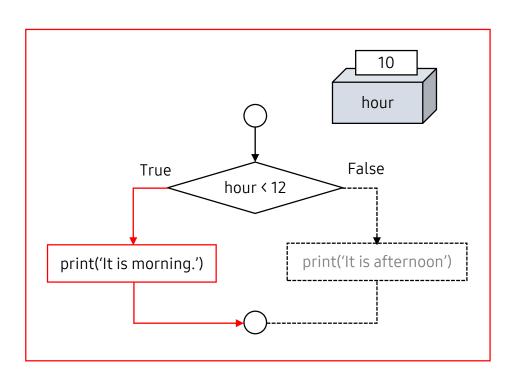
l ຮູບລຸ່ມນີ້ແມ່ນແຜນຜັງຂອງ Code ນີ້. hour False hour < 12 , True print('It is morning.') hour = 10if hour < 12: False hour>= 12 print('It is morning.') if hour >= 12: True print('It is afternoon.') print('It is afternoon.' It is morning.

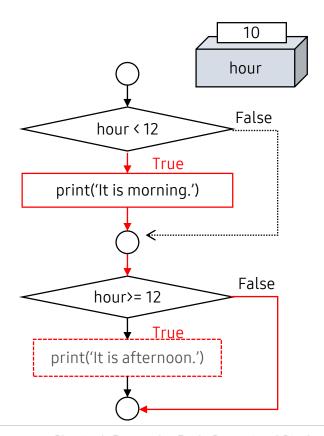
- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.3. ຄຳສັງ if-else (if else statement)
- l Code ກ່ອນໜ້ານີ້ ທີ່ໄດ້ໃຊ້ຄຳສັ່ງ if, ສາມາດຂຽນດ້ວຍຄຳສັ່ງ if-else ໄດ້ດັ່ງຕໍ່ໄປນີ້.
 - ມັນບໍ່ສາມາດເປັນຕອນເຊົ້າ ແລະ ຕອນບ່າຍພ້ອມກັນໄດ້. ຄວາມສຳພັນແບບນີ້ເອີ້ນວ່າຄວາມສຳພັນແບບພິເສດ.
 - ຄຳສັ່ງ if-else ເປັນຄຳສັ່ງທີ່ສາມາດໃຊ້ຄວາມສຳພັນແບບພິເສດດັ່ງກ່າວໄດ້.
 - ການສະແດງຄວາມສຳພັນແບບພິເສດກັບ if-else ເຮັດໃຫ້ແຜນຜັງງ່າຍຂຶ້ນ ສະແດງດັ່ງລຸ່ມນີ້.

```
hour = 10
    if hour < 12:
        print('It is morning.')
        print('It is afternoon.')
                                                                                                    hour
It is morning.
                                                                                                  False
                                                                         True
                                                                                   hour < 12
                                                               print('It is morning.')
                                                                                             print('It is afternoon'
```

- ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.4. ການປຸງບທຸງບແຜນຜັງຂອງ if-else ແລະ if
- lປູງບທຸງບແຜນຜັງຂອງ if-else ກັບການໃຊ້ if ສອງຄັ້ງ

ທາງເບື້ອງຊ້າຍຈະສັດເຈນກວ່າຫຼາຍ





- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.5. ຄຳສັ່ງ if-else ເພື່ອກຳນົດຕົວເລກຄີກ/ຄູ່
- I ກຳນົດຕົວເລກຄີກ/ຄູ່ດ້ວຍຄຳສັ່ງ if-else. (ໃນກໍລະນີຂອງຄວາມສຳພັນແບບພິເສດ, ສາມາດນຳໃຊ້ຄຳ ສັ່ງ if-else ໄດ້ດັ່ງຕໍ່ໄປນີ້)

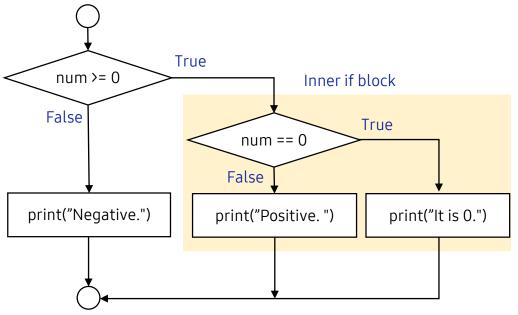
```
1  num = 10
2  if num % 2 == 0:
3     print(num, 'is an even number.')
4  else:
5     print(num, 'is an odd number.')
```

10 is and even number

Line 2, 4

- ຖ້າຕົວເລກຖືກຫານດ້ວຍ 2 ແລະ ສ່ວນເສດແມ່ນ 0, Block ແຖວທີ 3 ຈະຖືກປະຕິບັດ.
- ບໍ່ດັ່ງນັ້ນ Block ແຖວທີ 5 ຈະຖືກປະຕິບັດ.

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.6 ການຂຸງນຄຳສັ່ງ if-else ຢູ່ພາຍໃນຄຳສັ່ງ if-else



- ຖ້າຈຳເປັນອາດຈະລວມຄຳສັ່ງ if ອື່ນໄວ້ພາຍ
 ໃນຄຳສັ່ງ if. ນີ້ເອີ້ນວ່າຄຳສັ່ງ double if
- ຄຳສັ່ງ double if ສາມາດເກີດຂຶ້ນໄດ້ໃນ ສະຖານະການທີ່ເງື່ອນໄຂສະເພາະເພີ່ມເຕີມ ຕ້ອງການກວດສອບຫຼັງຈາກເງື່ອນໄຂທຳອິດ ມີຄ່າຄວາມຈິງເປັນ True.
- ໃນຕົວຢ່າງນີ້, ຫຼັງຈາກໄດ້ຮັບຕົວເລກຈາກຜູ້ໃຊ້, ຖ້າຄ່ານີ້ແມ່ນ num > = 0, ຈະມີການກວດ ສອບເງື່ອນໄຂອື່ນ ເພື່ອກວດເບິ່ງວ່າຕົວເລກ ດັ່ງກ່າວແມ່ນ 0 ຫຼື ບໍ່ ແລະ ພິມ "It is 0" ຫຼື "Positive".

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
 - 1.6. ການຂຸງນຄຳສັ່ງ if-else ຢູ່ພາຍໃນຄຳສັ່ງ if-else
 - l ພິຈາລະນາກໍລະນີທີ່ສັບຊ້ອນຂອງການນຳໃຊຄຳສັ່ງ if-else ພາຍໃນຄຳສັ່ງ if-else
 - ພວກເຮົາສາມາດແກ້ໄຂສະຖານະການຕ່າງໆດ້ວຍການ coding.
 - ສົມມຸດວ່າກໍລະນີທີ່ຜູ້ໃຊ້ໄດ້ຮັບຈຳນວນຖ້ວນ, ພິຈາລະນາວ່າຄ່ານັ້ນເປັນບວກ, ສູນ ຫຼື ລົບ ແລະ ພິມຜົນໄດ້ຮັບອອກທາງໜ້າຈໍ

```
1 | num = int(input("Enter an integer: "))
2 if num >= 0:  # make sure to indent the block
3 if num == 0:  # check specific conditions inside the block
           print("It is 0.")
    else:
           print("It is positive.")
7 else:
       print("It is negative.")
```

Enter an integer: -20 It is negative.

ດັ່ງນັ້ນ, ເຮົາສາມາດແກ້ໄຂບັນຫານີ້ໄດ້ໂດຍການໃຊ້ຄຳສັ່ງ if-else ພາຍໃນຄຳສັ່ງ if-else

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
 - 1.7 ການນຳໃຊ້ຄຳສັ່ງ if-else ທີ່ຊ້ອນກັນເພື່ອຈຳແນກຕົວເລກຄູ່/ຄີກ
 - l ໃຫ້ທົດລອງໃຊ້ if-else ອື່ນພາຍໃນ Block ທີ່ປະກອບດ້ວຍຄຳສັ່ງ if-else. ໃນ Code ນີ້, ເລກຄູ່ ແລະ ຄີກ ຈະຖືກກຳນົດ ແລະ ພິມສະເພາະໃນກໍລະນີຂອງຕົວເລກບວກເທົ່ານັ້ນ.

```
|num| = 100
2 if num < 0:
     print(num, 'is negative.')
     print(num, 'is not negative.')
   if num % 2 == 0:
         print(num, 'is an even number.')
     else:
         print(num, 'is an odd number.')
```

100 is not negative. 100 is even number.

Line 2, 4

• ເນື່ອງຈາກຕົວເລກໃຫຍ່ກວ່າ 0, Block ທາງຂວາດ້ານລຸ່ມຈະບໍ່ຖືກປະຕິບັດ, ແຕ່ຄຳສັ່ງ else ຂອງແຖວທີ 4 ຈະຖືກປະຕິບັດ.

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
 - 1.7 ການນຳໃຊ້ຄຳສັ່ງ if-else ຊ້ອນກັນ ເພື່ອຈຳແນກຕົວເລກຄູ່/ຄີກ

```
num = 100
2 if num < 0:
     print(num, 'is negative.')
 else:
     print(num, 'is not negative.')
    if num % 2 == 0:
        print(num, 'is an even number.')
     else:
        print(num, 'is an odd number.')
```

100 is not negative. 100 is an even number.

```
Line 5~7
```

- ຫຼັງຈາກການພິມ 'It is not negative' ໂດຍການນໍາໃຊ້ຟັງຊັນ print(), ໃຫ້ນໍາຕົວເລກມາ ຫານໃຫ້ 2 ອີກເທື່ອໜຶ່ງ ເພື່ອສົມທຸງບວ່າຄ່າເສດແມ່ນ 0 ຫຼື ບໍ່.
- ເນື່ອງຈາກ 100 % 2 ຄ່າເສດແມ່ນ 0, ຄຳສັ່ງ if ຈະຖືກປະຕິບັດ ແລະ ໂປຣແກຣມຈະຖືກ ຍົກເລີກ.

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
 - 1.8 ແຜນຜັງການຄວບຄຸມການເຮັດວຸງກໃນກໍລະນີຕ່າງໆ
 - l ລຸ່ມນີ້ແມ່ນກໍລະນີທີ່ num=-100.

```
num = -100
2 if num < 0:</pre>
      print(num, 'is negative.')
4 else:
      print(num, 'is not negative.')
     if num % 2 == 0:
          print(num, 'is an even number.')
      else:
          print(num, 'is an odd number.')
```

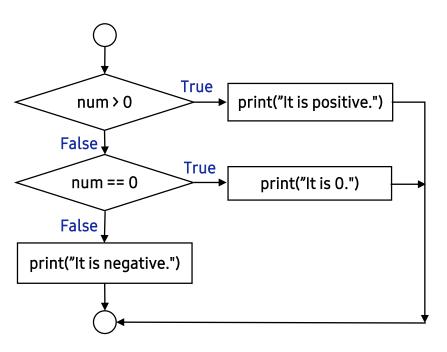
- -100 is negative.
- ໃນ code ນີ້ຈະນຳໃຊ້ if-else ສອງຄັ້ງ. ຄຳສັ່ງ if-else ຢູ່ທາງນອກ ເອີ້ນວ່າຄຳສັ່ງ if-else ພາຍນອກ ແລະ ຄຳສັ່ງ if-else ຢູ່ພາຍໃນ ເອີ້ນວ່າ ຄຳສັ່ງ if-else ພາຍໃນ. ໃຫ້ກວດເບິ່ງ code ຢ່າງລະອຽດ.
- ຄຳສັ່ງເງື່ອນໄຂ if-else ຂອງແຖວທີ 2 ແລະ 4 ແມ່ນຄຳສັ່ງ if-else ພາຍນອກ. ຄຳສັ່ງ if ຂອງແຖວ ທີ 2 ຈະຖືກປະຕິບັດກໍຕໍ່ເມື່ອຄ່າຂອງ num ນ້ອຍກວ່າ 0.

- 1. ການຕັດສິນໃຈແບບສອງທິດທາງ, ໂດຍໃຊ້ຄຳສັ່ງ else ແລະ elif
- 1.8. ແຜນຜັງການຄວບຄຸມການເຮັດວຸງກໃນກໍລະນີຕ່າງໆ
 - l ຂ້າງລຸ່ມນີ້ແມ່ນກໍລະນີທີ່ num=-100

```
1 |num = -100|
2 if num < 0:
      print(num, ''is negative.')
4 else:
     print(num, ''is not negative.')
  if num % 2 == 0:
         print(num, ''is an even number.')
         print(num, ''is an odd number.')
```

- -100 is negative.
- ຄຳສັ່ງ else ໃນແຖວທີ 4 ຈະຖືກປະຕິບັດກໍ່ຕໍ່ເມື່ອຄ່າຂອງ num ບໍ່ແມ່ນຄ່າລົບ.
- ແຖວທີ 6 ~ 9 ເປັນນຄຳສັ່ງ if-else ພາຍໃນຄຳສັ່ງ else ແລະ ເອີ້ນວ່າຄຳສັ່ງ if-else ພາຍໃນ.
- ຄຳສັ່ງ if ຂອງແຖວທີ 6 ຈະຖືກປະຕິບັດກໍ່ຕໍ່ເມື່ອ num ຫານດ້ວຍ 2 ໄດ້ຄ່າເສດເປັນ 0, ສະແດງວ່າເປັນເລກຄູ່.
- ລອງເບິ່ງແຖວທີ 8. else Block ນີ້ຈະຖືກປະຕິບັດເມື່ອເປັນເລກຄີກ.
- ດັ່ງນັ້ນ, ຖ້າຄ່າຂອງຕົວປຸ່ງນ num ຖືກປຸ່ງນເປັນ -100, ມັນຈະຖືກດຳເນີນການຕາມຂ້າງເທິງ.

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - 2.1 ວິທີການກວດສອບເງື່ອນໄຂອື່ນຢ່າງຕໍ່ເນື່ອງ ເມື່ອເງື່ອນໄຂບໍ່ເປັນຈິງ
 - l ວິທີການກວດສອບເງື່ອນໄຂອື່ນຢ່າງຕໍ່ເນື່ອງ ເມື່ອເງື່ອນໄຂບໍ່ເປັນຈິງ



- ພິຈາລະນາໂປຣແກຣມທີ່ຮັບຈຳນວນຖ້ວນຈາກຜູ້ໃຊ້ ແລະ ພິມ "It is positive.", "It is 0." ແລະ "It is negative."
- ຖ້າ True ຖືກສົ່ງຄືນຈາກຄຳສັ່ງ if ທຳອິດ, code ຈະຖືກ ປະຕິບັດ ແລະ ອອກຈາກຄຳສັ່ງເງື່ອນໄຂ.
- ຖ້າເປັນ False, ອາດຕ້ອງຕັດສິນໃຈຕາມເງື່ອນໄຂທີສອງ.
- ໂປຣແກຣມທີ່ມີໂຄງສ້າງດັ່ງກ່າວອາດຈະຖືກສ້າງໂດຍໃຊ້ if-elif-else.
- ▶ elif ເປັນຕົວຫຍໍ້ຂອງ "else if".

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
- 2.2 ຕົວຢ່າງ Code ທີ່ພິມ "It is positive.", "It is 0." ແລະ "It is negative." ຫຼັງຈາກໄດ້ ຮັບຈຳນວນຖ້ວນທີ່ເປັນ input.
- l ຂ້າງລຸ່ມນີ້ແມ່ນ code ທີ່ຮັບຈຳນວນຖ້ວນ, ໂດຍໃຊ້ຄຳສັ່ງ if-elif ແລະ ພິມ "It is positive.", "It is 0." ແລະ "It is negative."

```
1 | num = int(input ("Enter an integer: "))
3 if num > 0:
    print("It is positive.")
5 | elif num == 0:
     print("It is 0.")
 else:
     print("It is negative.")
```

```
Enter an integer: 6
It is positive.
Enter an integer: -5
It is negative.
Enter an integer: 0
It is 0.
```

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - 2.3 ເຫດຜົນໃນການໃຊ້ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - l ສົມມຸດບັນຫາການຕັດ Grade ທີ່ຈັດແບ່ງປະເພດຕາມລະດັບຂອງຄະແນນ, ໂດຍໃຊ້ຄຳສັ່ງ if ຫຼາຍຄັ້ງ. ຈະພິມຜົນການຮຽນ (Grades) ຕາມຊ້ວງຄະແນນ ທີ່ສະແດງໃນຕາຕະລາງລຸ່ມນີ້ໄດ້ແນວໃດ?
 - l ບັນຫາເຫຼົ່ານີ້ສາມາດແກ້ໄຂໄດ້ໂດຍໃຊ້ຄຳສັ່ງ if ແລະ ເງື່ອນໄຂພາຍໃນຄຳສັ່ງ if ຫຼາຍໆຄັ້ງ

Score	Grade
90 ~ 100	A
80 to Less than 90	В
70 to Less than 80	С
60 to Less than 70	D
Below 60	F

2.4 ຕົວຢ່າງການຕັດ Grade A: ນຳໃຊ້ Code ຄຳສັງ if ຫຼາຍໆຄັງ

```
1 | score = int(input('Enter score : '))
2 | if score >= 90:
                      # If 90 or above, 'A'
      grade = 'A'
4 if score < 90 and score >= 80: # If under 90, 80 or above, 'B'
      grade = 'B'
6 if score < 80 and score >= 70: # If under 80, 70 or above, 'C'
8 if score < 70 and score >= 60: # If under 70, 60 or above, 'D'
      grade = 'D'
                            #If under 60, 'F'
10 if score < 60:
      grade = 'F'
12 print ('Your grade is: ', grade)
```

- ຄະແນນ 88 ໄດ້ Grade B ແມ່ນຖືກຕ້ອງ
- ▶ ເຖິງຢ່າງໃດກໍຕາມ, code ນີ້ມີຄວາມສັບຊ້ອນຫຼາຍ ແລະ ອ່ານຍາກກ່ວາຄຳສັ່ງ if ທີ່ຜ່ານມາ.
- ເຖິງແມ່ນວ່າຈະມີສຳນວນເງື່ອນໄຂທີ່ບໍ່ຖືກຕ້ອງພາຍໃນແຕ່ລະເງື່ອນໄຂ, ຄວາມຜິດພາດນັ້ນຈະລະບຸໄດ້ຍາກຫຼາຍ.
- ຄວາມເປັນໄປໄດ້ຂອງຄວາມຜິດພາດເພີ່ມຂຶ້ນ ຍ້ອນວ່າຕ້ອງລະບຸຄວາມໝາຍຂອງຄຳສັ່ງ if ແຕ່ລະຄຳສັ່ງ.
- ເພື່ອແກ້ໄຂບັນຫານີ້, ໃຫ້ນຳໃຊ້ຄຳສັ່ງ if-else ດັ່ງຕໍ່ໄປນີ້

2.5 ຕົວຢ່າງການຕັດ Grade B: ນຳໃຊ້ Code ຄຳສັງ if-else

```
| score = int(input('Enter score : '))
2 | if score >= 90:
                               # If 90 or above, 'A'
      grade = 'A'
  else:
                             # If under 90, 80 or above, 'B'
      if score >= 80:
          grade = 'B'
      else:
         if score >= 70:
                          # If under 80, 70 or above, 'C'
             grade = 'C'
       else:
10
11
             if score >= 60:
                              # If under 70, 60 or above, 'D'
12
                 grade = 'D'
13
             else:
                                #If under 60. 'F'
                 grade = 'F'
15 print('Your grade is: ', grade)
```

- ຄະແນນ 88 ໄດ້ Grade B ແມ່ນຖືກຕ້ອງ.
- ໂອກາດທີ່ຈະເກີດຄວາມຜິດພາດຫຼຸດລົງ.
- ຄວາມສາມາດໃນການອ່ານແມ່ນຍັງຕໍ່າ ເພາະວ່າ if-else ສາມາດສະແດງໄດ້ພຽງແຕ່ສອງເງື່ອນໄຂເທົ່ານັ້ນ.

2.5 ຕົວຢ່າງການຕັດ Grade B: ນຳໃຊ້ Code ຄຳສັງ if-else

```
score = int(input('Enter score : '))
2 | if score >= 90:
                               # If 90 or above, 'A'
      grade = 'A'
  else:
                            # If under 90, 80 or above, 'B'
      if score >= 80:
          grade = 'B'
      else:
         if score >= 70:
                         # If under 80, 70 or above, 'C'
             grade = 'C'
9
10
       else:
11
             if score >= 60:
                              # If under 70, 60 or above, 'D'
                grade = 'D'
12
13
             else:
                               #If under 60. 'F'
                 grade = 'F'
15 print('Your grade is: ', grade)
```

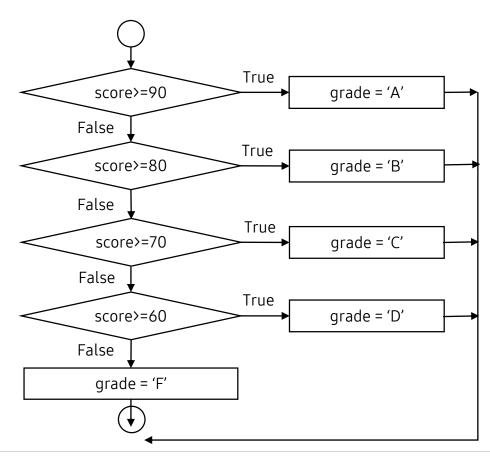
- มิทามพย้ำพ้าพายเทิมไป
- ສຳລັບຫຼາຍເງື່ອນໄຂ, ວິທີໃດທີ່ຈະເປັນວິທີທີ່ງ່າຍກວ່າ?

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - 2.6 ປຸງບທຸງບຄຳສັ່ງ if-elif ແລະ ຄຳສັ່ງ elif.

```
Code using if-elif
   Code using if-else
if score \geq 90:
                                            if score \geq 90:
     statement
                                                 statement
                                            elif score >= 80 :
else:
    if score >= 80 :
                                                 statement
        statement
                                            else:
    else:
                                                 statement
        statement
```

- codes ທັງສອງເຮັດວຸງກເໝືອນກັນ.
- code ດ້ານຂວາຖືກຫຍໍ້ໜ້າໜ້ອຍລົງ ແລະ ຈຳນວນຂອງແຖວຫຼຸດລົງ. ດັ່ງນັ້ນ code ຈຶ່ງອ່ານ ແລະ ທຳຄວາມເຂົ້າໃຈໄດ້ງ່າຍ.

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - 2.7 ຕົວຢ່າງຕົວຕັດ Grade C: ແຜນຜັງດຳເນີນການຂອງ if-elif-else statement



- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
 - 2.8 ຕົວຢ່າງການຕັດ Grade C: ນຳໃຊ້ Code ຄຳສັ່ງ if-elif-else

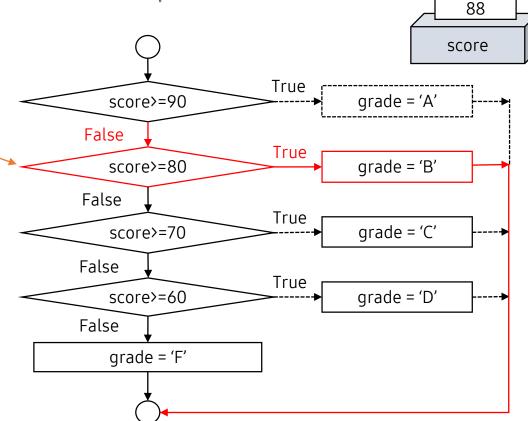
```
1 | score = int(input('Enter score : '))
2 if score >= 90: # If 90 or above, 'A'
      grade = 'A'
                    # If not 'A', 'B' if 80 or above
4 | elif score >= 80:
      grade = 'B'
6 elif score >= 70: # If not 'B', 'C' if 70 or above
      grade = 'C'
8 elif score >= 60: # If not 'C', 'D' if 60 or above
      grade = 'D'
                    # Otherwise, 'F'
10 else:
      grade = 'F'
12 print('Your grade is: ', grade)
```

- ຄະແນນ 88 ໄດ້ Grade B ແມ່ນຖືກຕ້ອງ.
- ▶ ເມື່ອປຸງບທຸງບກັບຕົວຢ່າງ A ແລະ B, ໂອກາດທີ່ຈະເກີດຄວາມຜິດພາດມີໜ້ອຍກວ່າ ເນື່ອງຈາກຕົວຢ່າງ C ໃຊ້ຄຳສັງ if-elif-else.
- ພ້ອມທັງ code ກໍເຂົ້າໃຈງ່າຍເພາະວ່າເງື່ອນໄຂສັດເຈນ.

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
- 2.8 ຕົວຢ່າງການຕັດ Grade C: ນຳໃຊ້ Code ຄຳສັ່ງ if-elif-else

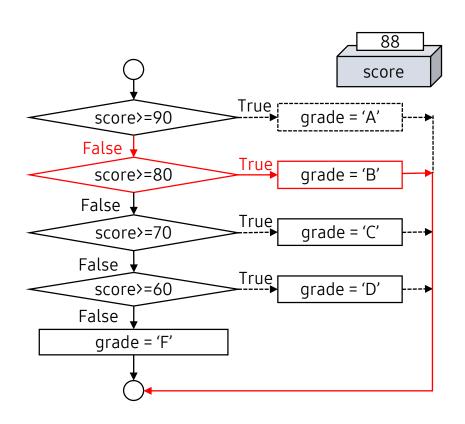
l ຕົວຢ່າງການຕັດ Grade C: Grade ຈະເປັນແນວໃດຖ້າວ່າຄະແນນໄດ້ 88?

ເມື່ອຄ່າຂອງຕົວປຸ່ງນຄະ ແນນແມ່ນ 88, ຈະມີຂັ້ນ ຕອນການປະຕິບັດຕໍ່ໄປນີ້



- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
- 2.8 ຕົວຢ່າງການຕັດ Grade C: ນຳໃຊ້ Code ຄຳສັ່ງ if-elif-else

l ຕົວຢ່າງການຕັດ Grade C: Grade ຈະເປັນແນວໃດ ຖ້າວ່າຄະແນນແມ່ນ 88? ປະຕິບັດຕາມແຜນຜັງ.



- ຂັ້ນຕອນທຳອິດ, ໂປຣແກຣມຈະປະຕິບັດຄຳສັ່ງ if score>=90: ເພື່ອພິຈາລະນາວ່າຄະແນນຫຼາຍກ່ວາ ຫຼື ເທົ່າ ກັບ 90 ບໍ່.
- ▶ ເນື່ອງຈາກຜົນຮັບຈາກການປຸງບທຸງບເປັນຄ່າ False, ຈຶ່ງດຳ ເນີນຕາມຄຳສັ່ງ elif score>=80: ເພື່ອພິຈາລະນາວ່າ ຄະແນນໃຫຍ່ກວ່າ ຫຼື ເທົ່າກັບ 80 ບໍ່.
- ▶ ເມື່ອຜົນໄດ້ຮັບນີ້ແມ່ນຄວາມຈິງເປັນ True, ປະໂຫຍກ grade = 'B' ຈະຖືກດຳເນີນການ ແລະ ການປະຕິບັດຄຳສັ່ງທີ່ elifelse ທັງໝົດຂ້າງລຸ່ມຈະຖືກຂ້າມໄປ.
- ດັ່ງນັ້ນ, ໃນທີ່ສຸດໂປຣແກຣມກໍຈະພິມ "your grade is : B.

2.9 ບັນຫາໃນຊີວິດຈິງທີ່ຫຼາກຫຼາຍ ແລະ ສັບຊ້ອນ



- ນັກຂຽນ ໂປຣແກຣມຕ້ອງຈັດການກັບບັນຫາທີ່ສັບຊ້ອນຫຼາຍກວ່ານີ້.
- 🗸 ຫາກຕ້ອງການກຳນົດເງື່ອນໄຂທີ່ຊັບຊ້ອນຫຼາຍຂຶ້ນ, ກໍສາມາດລວມ operators ການປຸງບທຸງບ ແລະ operators ດ້ານ logical ເຂົ້າດ້ວຍກັນ.
- ✔ Python ມີຕົວດຳເນີນການປຸງບທຸງບ ແລະ ຕົວດຳເນີນການ logical ດັ່ງຕໍ່ໄປນີ້, ທັງໝົດຈະຄືນຄ່າເປັນ ຂໍ້ມູນປະເພດ Boolean ຄ່າ ຖືກ, ຜິດ (True, True)
- ✓ ບາງຕົວຢ່າງເປັນການນຳໃຊ້ຕົວດຳເນີນການຫຼາຍອັນລວມກັນ.

Comparison operator	Explanation
==	Returns True if two operands have identical value.
!=	Returns True if the values of the two operands are different.
>	Returns True if the left operand is greater than the right operand.
<	Returns True if the left operand is less than the right operand.
>=	Returns True if the left operand is greater than or equal to the right operand.
<=	Returns True if the left operand is less than or equal to the right operand.

Logical operator	Meaning
x and y	If one of x or y is False, it is False. It is only True if it all are True.
x or y	If either x or y is True, it is True, and False only if all are False.
not x	If x is true, it is False, and if x is false, it is True.

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
- 2.10 ຕົວຢ່າງຂອງສຳນວນທີ່ມີເງື່ອນໄຂສັບສົນ

```
# if b is changed to 13, it does not meet the first conditional statement
3 if (a \% 2 == 0) and (b \% 2 == 0):
                                   # first conditional statement
     print('Both are positive.')
5 if (a \% 2 == 0) or (b \% 2 == 0):
                                  # second conditional statement
     print('At least one is positive.')
```

Both are positive. At least one is positive.

- code ຂ້າງເທິງທີ່ໄດ້ຜົນຮັບດ້ວຍສອງເງື່ອນໄຂເມື່ອ a = 10 ແລະ b = 14
- ຄຳສັ່ງ if ໃນ code ນີ້ຈະພິມຜົນຮັບອອກມາ, ໂດຍການໃຊ້ຕົວດຳເນີນການທາງຕັກກະສາດຄື and ແລະ or.
- ▶ ໃນ code ນີ້ (a % 2 == 0) ແມ່ນ True ແລະ (b % 2 == 0) ແມ່ນ True. ດັ່ງນັ້ນຈຶ່ງ ພິມຄຳສັ່ງທັງສອງຄຳສັ່ງຂອງແຖວ 4 ແລະ 6 ອອກມາ.

- 2. ຄຳສັ່ງທີ່ມີເງື່ອນໄຂຊ້ອນກັນ
- 2.10 ຕົວຢ່າງຂອງສຳນວນທີ່ມີເງື່ອນໄຂສັບສົນ

```
# if b is changed to 13, it does not meet the first conditional statement
3 if (a % 2 == 0) and (b % 2 == 0): # first conditional statement
4 print ('Both numbers are even..')
5 if (a \% 2 == 0) or (b \% 2 == 0):
                                # second conditional statement
     print('More than one is positive.')
```

More than one is positive.

- code ຂ້າງເທິງທີ່ສ້າງຜົນຮັບຈາກສອງເງື່ອນໄຂເມື່ອ a = 10 ແລະ b = 13.
- ຄຳສັ່ງ if ໃນ code ນີ້ຈະພິມຜົນຮັບອອກມາ, ໂດຍການໃຊ້ຕົວດຳເນີນການທາງຕັກກະສາດຄື and ແລະ or.
- ໃນທີ່ນີ້ເນື່ອງຈາກວ່າ b ມີຄ່າເທົ່າກັບ 13, (b % 2 == 0) ມີຄ່າເປັນ False.
- ▶ ດັ່ງນັ້ນ (a % 2 == 0) and (b % 2 == 0) ເປັນ False, ສະນັ້ນ 'Both numbers are even.' ບໍ່ໄດ້ພິມອອກມາ.
- ▶ ໃນທາງກົງກັນຂ້າມ, ເນື່ອງຈາກ (a % 2 == 0) ເປັນ True, ດັ່ງນັ້ນ (a % 2 ==0) or (b % 2 == 0) ກໍເປັນ True.

One More Step

Code ທີ່ງ່າຍກວ່າ, ການຂຽນ code ກໍງ່າຍຂຶ້ນ ແລະ ຜູ້ອ່ານກໍເຂົ້າໃຈງ່າຍຂຶ້ນ. Code ດັ່ງກ່າວ ເອີ້ນວ່າ "codes with good readability." ຫາກຕ້ອງການກວດສອບພາລະກິດທີ່ຜ່ານມາ, ໃຫ້ ຊອກຫາ leap year ໂດຍການໃຊ້ສຳນວນທີ່ມີເງື່ອນໄຂຊັບຊ້ອນ ແລະ ກຳນົດໃຫ້ກັບ is_leap_year. is_leap_year ແມ່ນ True ເມື່ອຄ່າປີເປັນ leap year, ຖ້າບໍ່ດັ່ງນັ້ນກໍເປັນ False.

```
1 year = int(input('Enter the year: '))
2 is leap year = ((year \% 4 == 0)) and (year \% 100 != 0) or (year \% 400 == 0))
3 print(year, 'is a leap year?is_leap year)
```

Enter the year: 2021 2021 is a leap year? False

ຖ້າໃຊ້ສຳນວນທີ່ມີເງື່ອນໄຂທີ່ຊັບຊ້ອນໄດ້ດີ, code ກໍຈະມີຄວາມຊັດເຈນຫຼາຍຂຶ້ນ.

- 3. ແນວທາງປະຕິບັດໃນການສ້າງ Application
- 3.1. ແບບເຝິກຫັດ Application
- l ສົມມຸດວ່າມີໝາກໄມ້ 4 ຊະນິດຢູ່ໃນຮ້ານໝາກໄມ້ຂອງ David ຕາມຮູບຂ້າງລຸ່ມນີ້ ແລະ ໃຫ້ປ້ອນຊະນິດ ແລະ ຈຳນວນໝາກໄມ້ເຂົ້າໄປໃນລະບົບ.
- l ເຮັດໃຫ້ code ພິມຂໍ້ຄວາມແຈ້ງໃຫ້ຜູ້ໃຊ້ຮູ້ຈຳນວນເງິນທັງໝົດ ແລະ ຂໍເງິນຄ່າສິນຄ້າ.
- l ພິມຂໍ້ມູນສິນຄ້າ ແລະ ບອກຈຳນວນເງິນທອນ. ຖ້າເງິນຍັງບໍ່ພຸງພໍ ໃຫ້ສົ່ງຂໍ້ຄວາມບອກວ່າເງິນບໍ່ພຸງໝໍ.

```
This is David's fruit shop.
1: Apple(price: 5000 won)
2: Grape(price: 6000 won)
3: Melon(price: 8000 won)
4: Orange(price: 2000 won)
Enter item number (between 1\sim4) >> 2
Enter number of item(between 1\sim10) >> 2
Fruit selected : grape
Price : 6000
Quantity: 2
Total price is 12000 won.
Insert money please(ex: 15000) >>> 20000
20000 won received. Your change is 8000 won.
```

- 3. ແນວທາງປະຕິບັດໃນການສ້າງ Application
- 3.1. ແບບເຝິກຫັດ Application

l ສະແດງເມນູ ແລະ ຮັບຂໍ້ມູນດັ່ງຕໍ່ໄປນີ້.

```
1 print('#' * 50)
print('This is David's fruit shop.')
3 print('1: Apple( price : 5000 won)')
4 print('2: Grape( price : 6000 won)')
5 print('3: Melon( price : 8000 won)')
6 print('4: Orange( price : 2000 won)')
  print('#' * 50)
9 total_price = 0
10 order = int(input(('Enter item number(between 1~4) >> ')))
11 | count = int(input(('Enter number of items(between 1~10) >> ')))
12
```

- 3. ແນວທາງປະຕິບັດໃນການສ້າງ Application
- 3.1. ແບບເຝິກຫັດ Application

l ກຳນົດໝາກໄມ້ທີ່ເລືອກ, ລາຄາ ແລະ ຄິດໄລ່ລາຄາທັງໝົດ. ໃຊ້ຄຳສັ່ງ if ເພື່ອພິມ 'Not enough money.' ຖ້າເງິນບໍ່ພຸງງພໍ. ຖ້າບໍ່ດັ່ງນັ້ນກໍໃຫ້ພິມຈຳນວນເງິນທອນ.

```
13 | if order == 1:
      fruit = 'Apple'
      price = 5000
16 | elif order == 2:
     fruit = 'Grape'
      price = 6000
19 | elif order == 3:
      fruit = 'Melon'
20
       price = 8000
22 else:
23
      fruit = 'Orange'
24
       price = 2000
25
26 print('Fruit selected :', fruit)
27 print('Price :', price)
28 print('Quantity:', count)
29 print('Total price is', price * count, 'won.')
30
31 money = int(input('Insert money please(ex: 15000) >>>'))
32 | if money < price * count :
       print('Not enough money.')
34 else:
35
       change = money - price * count
       print(money, 'won received. Your change is', change, 'won.')
```

- 3. ແນວທາງປະຕິບັດໃນການສ້າງ Application
- 3.2 ຄຳອະທິບາຍ code ທີ່ສຳຄັນຂອງຕົວຢ່າງ application
 - ຈຸດສຳຄັນຂອງໂປຣແກຣມແມ່ນດັ່ງຕໍ່ໄປນີ້
 - ຖ້າຈຳນວນຄຳສັ່ງຊື້ແມ່ນ 1, ໝາກໄມ້ຄື 'apple' ແລະ ລາຄາແມ່ນ 5000.
 - ກ້າບໍ່ແມ່ນ, code ຈະດຳເນີນການກວດສອບເງື່ອນໄຂອື່ນໂດຍຜ່ານຄຳສັ່ງ if-elif.
 - ດ້ວຍວິທີດັ່ງກ່າວ, ຄຳສັ່ງ if-elif-else ຈະເຮັດໃຫ້ code ງ່າຍຂຶ້ນ.

```
13 if order == 1:
14 fruit = 'Apple'
      price = 5000
16 | elif order == 2:
   fruit = 'Grape'
      price = 6000
19 elif order == 3:
   fruit = 'Melon'
21 price = 8000
22 else:
23
   fruit = 'Orange'
      price = 2000
24
```

Paper coding

- ✓ ພະຍາຍາມທຳຄວາມເຂົ້າໃຈແນວຄິດພື້ນຖານຢ່າງຖີ່ຖ້ວນ ກ່ອນທີ່ຈະກ້າວ ໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.
- ✓ ການບໍ່ມີຄວາມເຂົ້າໃຈແນວຄວາມຄິດພື້ນຖານ ຈະເພີ່ມພາລະໃນການຮຽນ ໃນຫຼັກສູດນີ້, ເຊິ່ງອາດຈະເຮັດໃຫ້ລົ້ມເຫຼວໃນຫຼັກສູດນີ້.
- ✓ ໃນຕອນນີ້ອາດຈະເປັນເລື່ອງຍາກ, ແຕ່ເພື່ອໃຫ້ສຳເລັດໃນຫຼັກສູດນີ້, ຂໍແນະ ນຳໃຫ້ເຂົ້າໃຈແນວຄວາມຄິດຢ່າງລະອຸງດ ແລະ ກ້າວໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.

ຮັບຕົວອັກສອນຈາກຜູ້ໃຊ້ ແລະ ພິມ 'It is a vowel' ສຳລັບ a, e, i, 0, u, ແລະ 'It is a Q1. consonant' ສຳລັບຕົວອັກສອນອື່ນໆ.

Example Output	Enter the alphabet : k K is a consonant
Time	5min

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

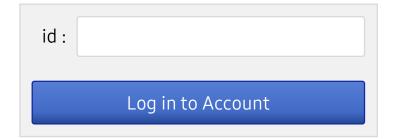
ຂງນໂປຣແກຣມຕໍ່ໄປນີ້, ເຊິ່ງຮັບຂໍ້ມູນຈຳນວນຖ້ວນ a, b ສອງຕົວ, ກຳນົດວ່າ a ເປັນຜົນຄູນຂອງ b ຫຼື Q2. ບໍ່ ແລະ ພິມຜົນໄດ້ຮັບອອກມາ.

Example Output	Write two integers: 30 3 30 is a multiple of 3
Time	5min

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

Let's code

- 1. ຕົວຢ່າງການນຳໃຊ້ if-elif
- **1.1** ໂປຣແກຣມສໍາລັບກວດສອບ ID
- l ໃຫ້ຂຽນໂປຣແກຣມທີ່ຮັບ ID ຈາກຜູ້ໃຊ້ ແລະ ພິມວ່າມັນກົງກັບ ID 'ilovepython' ທີ່ເກັບໄວ້ໃນໂປຣ ແກຣມ ຫຼື ບໍ່.



- 1. ຕົວຢ່າງການນໍາໃຊ້ if-elif
- **1.2.** code ຂອງໂປຣແກຣມສໍາລັບການກວດສອບ ID.

l code ການແກ້ໄຂບັນຫາໂດຍໃຊ້ຄຳສັ່ງ if-else

```
d = "ilovepython"
s = input("Enter the ID: ")
if s == id:
    print("Welcome.")
else:
    print("ID not found.")
```

Enter the ID: ilovepython Welcome.

ຮັບຂໍ້ມູນປ້ອນເຂົ້າເປັນ string ໂດຍໃຊ້ຟັງຊັນການປ້ອນຂໍ້ມູນ, ຈາກນັ້ນຕົວດຳເນີນການແບບ
 ປງບທງບ == ຈະກວດສອບເບິ່ງວ່າ ປ້ອນຂໍ້ມູນ string ກົງກັບ 'ilovepython'. ຫຼື ບໍ່, ຖ້າຜົນໄດ້ ຮັບເປັນ True, ໃຫ້ພິມ ' Welcome '.

One More Step

ກວດສອບລະຫັດຜ່ານຫຼັງຈາກການກວດສອບ ID ສຳເລັດ, ນັ້ນຄືການຂຸງນໂປຣ ແກຣມດ້ວຍ output ຕໍ່ໄປນີ້, ເຊິ່ງລະຫັດຜ່ານທີ່ເກັບໄວ້ແມ່ນ mypass1234. ຖ້າທັງ ສອງຖືກຕ້ອງ, ໃຫ້ພິມ "Welcome". ຖ້າບໍ່ດັ່ງນັ້ນໃຫ້ພິມ "ID not found " ຫຼື "The password is wrong."

Enter your ID:	
Password	
Log in to Account	

Enter ID: Enter Password: welcome.

1. ຕົວຢ່າງການນໍາໃຊ້ if-elif

1.3. ໂປຣແກຣມກວດສອບ ID ແລະ ລະຫັດຜ່ານ

```
1 my_id = "ilovepython"
password = "mypass1234"
3 string1 = input("Enter ID: ")
4 string2 = input("Enter Password: ")
5 if string1 == my_id and string2 == password:
 print("Welcome.")
elif string1 != my_id:
      print("ID not found.")
  else:
      print("The password is wrong.")
```

Enter ID: ilovepython Enter password: mypass1234 welcome.

```
Line 2~3
   • ຮັບ ID ແລະ ລະຫັດຜ່ານຈາກຜູ້ໃຊ້.
```

- 1. ຕົວຢ່າງການນໍາໃຊ້ if-elif
- 1.3. ໂປຣແກຣມກວດສອບ ID ແລະ password

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter ID: ")
4 | string2 = input("Enter Password: ")
5 if string1 == my id and string2 == password:
      print("welcome.")
7 elif string1 != my id:
      print("ID not found.")
 else:
      print("The password is wrong.")
```

Enter ID: ilovepython Enter password: mypass1234 welcome.

```
্রাট্র Line
5~6
```

- ປຸງບທຸງບການປ້ອນຂໍ້ມູນ string ໂດຍຜູ້ໃຊ້ວ່າ ກົງກັບ my_id ແລະ ລະຫັດຜ່ານ ຫຼື ບໍ່. ເນື່ອງຈາກນຳ ໃຊ້ຕົວດຳເນີນການ and, Block ຂ້າງລຸ່ມນີ້ຈະບໍ່ຖືກປະຕິບັດ, ຖ້າຫາກວ່າອັນໃດອັນໜຶ່ງເປັນ False.
- ຖ້າແຖວທີ 5 ມີຄ່າເປັນ True, ກໍຈະພິມ 'Welcome' ອອກມາ.

1. ຕົວຢ່າງການນໍາໃຊ້ if-elif

1.3. ໂປຣແກຣມກວດສອບ ID ແລະ password

```
1 my id = "ilovepython"
password = "mypass1234"
3 string1 = input("Enter ID: ")
4 | string2 = input("Enter Password: ")
5 if string1 == my id and string2 == password:
      print("welcome.")
7 | elif string1 != my id:
      print("ID not found.")
9 else:
      print("The password is wrong.")
```

Enter ID: ilovepython Enter password: mypass1234 welcome.

```
III Line 7~8
```

- ຖ້າຄຳສັ່ງ if ໃນແຖວທີ 5 ມີຄ່າຄວາມຈິງເປັນ false, ກວດເບິ່ງວ່າ my id ແຕກຕ່າງຈາກ string (ຄ່າການປ້ອນຂໍ້ມູນຂອງຜູ້ໃຊ້) ໃນຄຳສັ່ງ elif ຫຼື ບໍ່.
- ຖ້າແຕກຕ່າງກັນ, ລະບົບຈະພິມ 'ID not found' ອອກມາ.

1. ຕົວຢ່າງການນໍາໃຊ້ if-elif

1.3. ໂປຣແກຣມກວດສອບ ID ແລະ password

```
1 my id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter ID: ")
4 string2 = input("Enter Password: ")
5 if string1 == my_id and string2 == password:
      print("welcome.")
7 | elif string1 != my id:
      print("ID not found.")
 else:
      print("The password is wrong.")
```

Enter ID: ilovepython Enter password: mypass1234 welcome.

```
ត់តែ Line 9~10
```

• ຖ້າທັງສອງຄຳສັງ if ແລະ elif ໃນແຖວທີ 5 ແລະ ແຖວທີ 7 ມີຄ່ຳຄວາມຈິງເປັນ false, ຄຳສັ່ງ print ຂອງຄຳສັ່ງ else ສຸດທ້າຍຈະຖືກດຳເນີນການ.

- 2.ຕົວຢ່າງ ໂມດູນແບບສູ່ມ (random)
- 2.1. ຕົວຢ່າງ Code ຂອງການສ້າງຕົວເລກແບບສຸ່ມ
 - l ສຶກສາການສ້າງຕົວເລກແບບສຸ່ມທີ່ຈຳເປັນ ເພື່ອແກ້ໄຂບັນຫາໃນພາລະກິດນີ້, ຕົວເລກ Random ສາມາດສ້າງຢູ່ໃນ Python ໄດ້ດັ່ງຕໍ່ໄປນີ້. code ຂ້າງລຸ່ມນີ້ສ້າງຕົວເລກສຸ່ມລະຫວ່າງ 1 ຫາ 100 l ເມື່ອເອີ້ນໃຊ້ຟັງຊັນ Random.randint(1, 100) ຂອງໂມດູນແບບສູ່ມ, ຈະມີການສ້າງຕົວເລກສູ່ມ ລະຫວ່າງ 1 ແລະ 100 (ຄ່າຕົວເລກສຸ່ມປະກອບດ້ວຍ 1 ຫາ 100).

```
# call the random module
import random
number = random.randint(1, 100)
print(number)
```

93

温温 Line 1

• ໃນການນຳໃຊ້ຟັງຊັນ randint ເພື່ອສ້າງຕົວເລກແບບສຸ່ມລະຫວ່າງ 1 ຫາ 100, ຈະຕ້ອງນຳ ເຂົ້າ ໂມດູນຕົວເລກແບບສຸ່ມ, ໂດຍໃຊ້ຄຳສັ່ງ import.

- 2. ຕົວຢ່າງ ໂມດູນແບບສຸ່ມ (random)
- 2.2. ເກມໂຍນຫຼຸງນໂດຍໃຊ້ຟັງຊັນ random

l ຂຸງນ code ເກມໂຍນຫຼຸງນ. ການໂຍນຫຼຸງນສາມາດເຮັດໄດ້ໂດຍການສ້າງຕົວເລກແບບສຸ່ມ.

l ໃນ Python, ຫຼັງຈາກທີ່ນຳ random ເຂົ້າມາ ແລະ ເອີ້ນໃຊ້ຟັງຊັນ random.randrange (2), ຕົວເລກ 0 ຫຼື 1 ຈະຖືກສ້າງຂຶ້ນແບບສຸ່ມ (random)





- 2. ຕົວຢ່າງ ໂມດູນແບບສຸ່ມ (random)
- 2.2. ເກມໂຍນຫຼຸງນໂດຍໃຊ້ຟັງຊັນ random

```
import random
number = random.randrange(2)
print(number)
```

1

- ນີ້ຄື code ທຶດສອບສໍາລັບສ້າງເກມໂຍນຫຼຸງນດ້ວຍຟັງຊັນ random.
- Run ໄດ້ຫຼາຍຄັ້ງ. ໂປຣແກຣມຈະພິມ 0 ຫຼື 1

2. random Module Preview

2.2. ເກມໂຍນຫຼຸງນໂດຍໃຊ້ຟັງຊັນ random

l ສຸ່ມສ້າງ 0 ຫຼື 1 ໂດຍໃຊ້ຟັງຊັນ randrange

l ຖ້າຄ່ານີ້ແມ່ນ 0, ໃຫ້ພິມ 'front' ດັ່ງທີ່ສະແດງຢູ່ໃນ code ລຸ່ມນີ້, ຖ້າບໍ່ດັ່ງນັ້ນໃຫ້ພິມ 'back'.

```
import random
3 print("Start coin tossing game. ")
4 coin = random.randrange(2)
5 | if coin == 0:
      print("Front.")
7 else:
      print("Back.")
  print("Game finished.")
```

Start coin tossing game. Front. Game finished.

```
Line 4
```

• ຟັງຊັນທີ່ເອີ້ນວ່າ Randrange(2) ເປັນຟັງຊັນທີ່ຄືນຄ່າ 0 ຫຼື 1 ຜ່ານຕົວສ້າງຕົວເລກແບບສຸ່ມ.

Pair programming



ແບບເຝິກຫັດການຂຸງນໂປຣແກຣມ



ແນວທາງ, ກົນໄກ ແລະ ແຜນສຸກເສີນ

ການກະກຽມໂປຣແກຣມຈັບຄູ່ກ່ຽວຂ້ອງກັບການກຳນົດແນວທາງ ແລະ ກົນໄກ ເພື່ອຊ່ວຍໃຫ້ນັກຮຽນຈັບຄູ່ໄດ້ຢ່າງຖືກຕ້ອງ ແລະ ເພື່ອໃຫ້ພວກເຂົາຈັບຄູ່ ກັນ. ຕົວຢ່າງເຊັ່ນ ນັກຮຽນຄວນຜັດປ່ຽນກັນ "ບັງຄັບ mouse" ການກະກຽມທີ່ມີປະສິດຕິພາບ ຈຳເປັນຕ້ອງໃຫ້ມີແຜນສຸກເສີນ ໃນກໍລະນີທີ່ຄູ່ຮ່ວມ ງານຄົນໃດຄົ້ນໜຶ່ງບໍ່ຢູ່ ຫຼື ຕັດສິນໃຈທີ່ຈະບໍ່ເຂົ້າຮ່ວມດ້ວຍເຫດຜົນໃດໜຶ່ງ ຫຼື ເຫດຜົນອື່ນ. ໃນກໍລະນີເຫຼົ່ານີ້, ສິ່ງສຳຄັນຄືຕ້ອງໄດ້ລະບຸໃຫ້ຊັດເຈນວ່າ ນັກຮຽນທີ່ມີການປະກອບສ່ວນຈະບໍ່ຖືກລົງໂທດຍ້ອນວ່າການຈັບຄູ່ບໍ່ໄດ້ຜົນດີ.

ການຈັບຄູ່ຄວາມສາມາດທີ່ຄ້າຍຄືກັນ, ບໍ່ຈຳເປັນຕ້ອງມີຄວາມສາມາດເທົ່າທຽມກັນ ໃນຖານະເປັນຄູ່ຮ່ວມງານກັນ

ການຂຽນໂປຣແກຣມຈັບຄູ່ ທີ່ມີປະສິດທິພາບເມື່ອນັກຮຽນທີ່ມີຄວາມສາມາດຄ້າຍຄືກັນ, ເຖິງວ່າບໍ່ຈຳເປັນຕ້ອງມີຄວາມເທົ່າທຽມກັນຖືກຈັບຄູ່ເປັນຄູ່ຮ່ວມ ງານກັນ. ການຈັບຄູ່ນັກຮຽນທີ່ບໍ່ກິງກັນມັກຈະເຮັດໃຫ້ການມີສ່ວນຮ່ວມທີ່ບໍ່ສົມດູນກັນ. ຄູສອນຕ້ອງເນັ້ນໜັກວ່າການຂຽນໂປຣແກຣມຈັບຄູ່ບໍ່ແມ່ນ ກົນລະຍຸດ "ແບ່ງປັນ ແລະ ເອົາຊະນະ", ແຕ່ເປັນຄວາມພະຍາຍາມຮ່ວມມືກັນທີ່ແທ້ຈິງ ໃນທຸກໆຄວາມພະຍາຍາມສໍາລັບໂຄງການທັງໝົດ. ຄູຄວນ ຫຼືກເວັ້ນການຈັບຄູ່ນັກຮຽນທີ່ອ່ອນຫຼາຍກັບນັກຮຽນທີ່ເກັ່ງຫຼາຍ..

ສ້າງແຮງຈູງໃຈໃຫ້ກັບນັກຮຽນໂດຍການສະເໜີສິ່ງຈູງໃຈເພີ່ມເຕີມ

ການສະເໜີແຮງຈູງໃຈເພີ່ມເຕີມສາມາດຊ່ວຍກະຕຸ້ນນັກຮຽນໃຫ້ຈັບຄູ່, ໂດຍສະເພາະກັບນັກຮຽນທີ່ຮຽນເກັ່ງ. ຄູບາງຄົນພົບວ່າການກຳນົດໃຫ້ນັກຮຽນຈັບ ຄູ່ພຽງແຕ່ໜຶ່ງ ຫຼື ສອງວຽກນັ້ນຈະມີຜົນດີ.



Pair Programming Practice



Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

ຈົ່ງຊຽນໂປຣແກຣມທີ່ນຳໃຊ້ການບວກ, ລົບ, ຄູນ ແລະ ຫານ. ໃຫ້ພິມຜົນຮັບຈາກການປະມວນຜົນຂອງຈຳນວນທຳມະ ຊາດສອງຕົວ, ຕາມຕົວເລກ ແລະ ຕົວດຳເນີນການທີ່ຕ້ອງການຈາກ input. ຖ້າປ້ອນຕົວເລກອື່ນທີ່ບໍ່ແມ່ນ 1, 2, 3 ແລະ Q 1.4 ສະແດງວ່າປ້ອນຕົວເລກບໍ່ຖືກຕ້ອງ, ຈະພິມ 'Entered an incorrect number' ອອກມາ. ໃນການປ້ອນຕົວເລກສອງ ຕົວ, ໃຫ້ພິມຕົວເລກທີໜຶ່ງ, ແລ້ວກົດ enter, ຈາກນັ້ນໃຫ້ພິມຕົວເລກຕົວທີ່ສອງ.

Output example

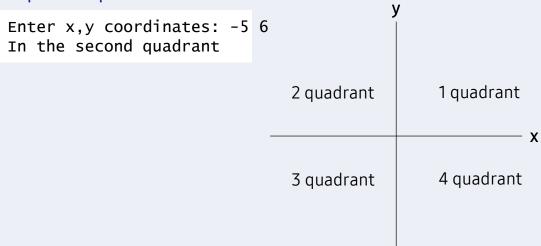
```
1) Addition 2) Subtraction 3) Multiplication
                                                   4) Division
Enter the desired number of operation: 1
Enter two numbers for operation.
10
20
10 + 20 = 30
```

If inserted incorrectly

```
1)Addition
              2) Subtraction
                              3) Multiplication
                                                 4) Division
Enter the desired number of operation: 5
Entered an incorrect number.
```

Q2. ໃຫ້ຂຽນໂປຣແກຣມທີ່ຮັບເມັດທີ່ມີຈຸດພິກັດ x ແລະ y ເປັນ input ແລະ ກຳນົດວ່າ ເມັດດັ່ງກ່າວຢູ່ໃນພື້ນທີ່ locaeງພື້ນທີ່ 1, 2, 3, 4. ຕຳແໜ່ງຂອງພື້ນທີ່ແມ່ນສະແດງຢູ່ໃນຮູບຕໍ່ໄປນີ້.

Output example



Q3. ຜູ້ໃຊ້ເລືອກໜຶ່ງລາຍການ. ຖ້າຕົວອັກສອນທີ່ປ້ອນເຂົ້າໄປບໍ່ຢູ່ໃນເມນູ, ໃຫ້ພິມ 'enter the menu again' ແລະ ຮັບການປ້ອນຂໍ້ມູນໃໝ່.

Output example

```
Welcome to yummy restaurant. Here is the menu.
- Burger(enter b)
- Chicken(enter c)
- Pizza(enter p)
Choose a menu (enter b,c,p) : b
You chose pizza.
```

 ສິ່ງນີ້ຕ້ອງການສຳນວນເງື່ອນໄຂທີ່ສັບຊ້ອນ. ລວມຕົວດຳເນີນການແບບຕັກກະສາດ ແລະ ຄຳສັ່ງ ທີ່ມີເງື່ອນໄຂຢ່າງລະມັດລະວັງ.