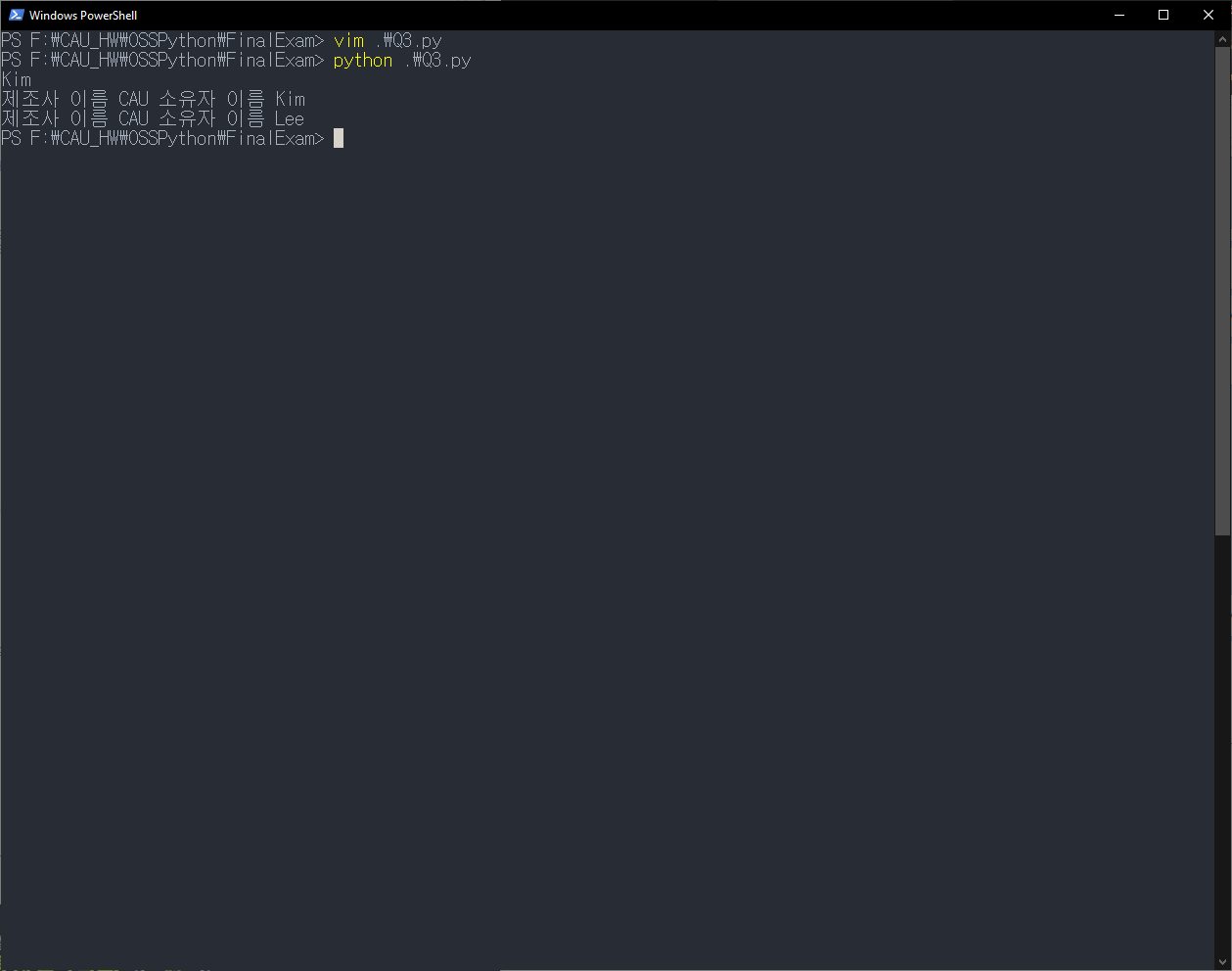
OSS finals(20216793 김준섭)

[Question-1 (4 points)]  
원본 저장소를 소스트리의 리모트 저장소에 추가한 후 원본 저장소에서 fetch를 해온다. 그러고 conflict가 발생할테니 conflict를 먼저 해결하고 commit을 진행한 후 pull request를 진행한다. 또는fetch를 한 remote repository의 commit으로 현재 local repository의 commit을 Rebase 를 하여 conflict를 해결하고 local repository를 push –force를 이용하여 remote repository에 올리고 이를 pull request를 진행한다.

[Question-2 (3 points)]  
amend : 방금 했던 커밋을 수정할 수 있다. 따라서 직전에 진행한 커밋이 잘못되었다면 수정할 수 있다.  
hard-reset : 로컬 저장소의 모든 작업상태 내의 변경사항을 버리고 이전 커밋의 상태로 돌아간다. 잘못된 커밋의 기록을 완전히 지우고 싶을 때 사용한다.  
revert : 리셋과 같이 변경사항을 모두 없애는 것이 아닌 이전의 커밋 상태를 되돌려 새로운 커밋으로써 추가하는 기능이다. 다수가 사용하는 환경에서 이력 관리가 중요할 때 reset보다 revert를 사용해서 이력이 남기며 돌아가는 것이 더 좋다.

[Question-3 (5 points)]  
  
class Television:

maker=""

def \_\_init\_\_(self, maker="CAU"):

Television.maker = maker

self.\_\_owner = ""

def getOwner(self):

return self.\_\_owner

def setOwner(self, owner):

self.\_\_owner = owner

def \_\_str\_\_(self):

return "제조사 이름 {} 소유자 이름 {}".format(Television.maker, self.\_\_owner)

myTv = Television("MM") # Assume that "MM" is the maker of the Television

yourTv = Television()

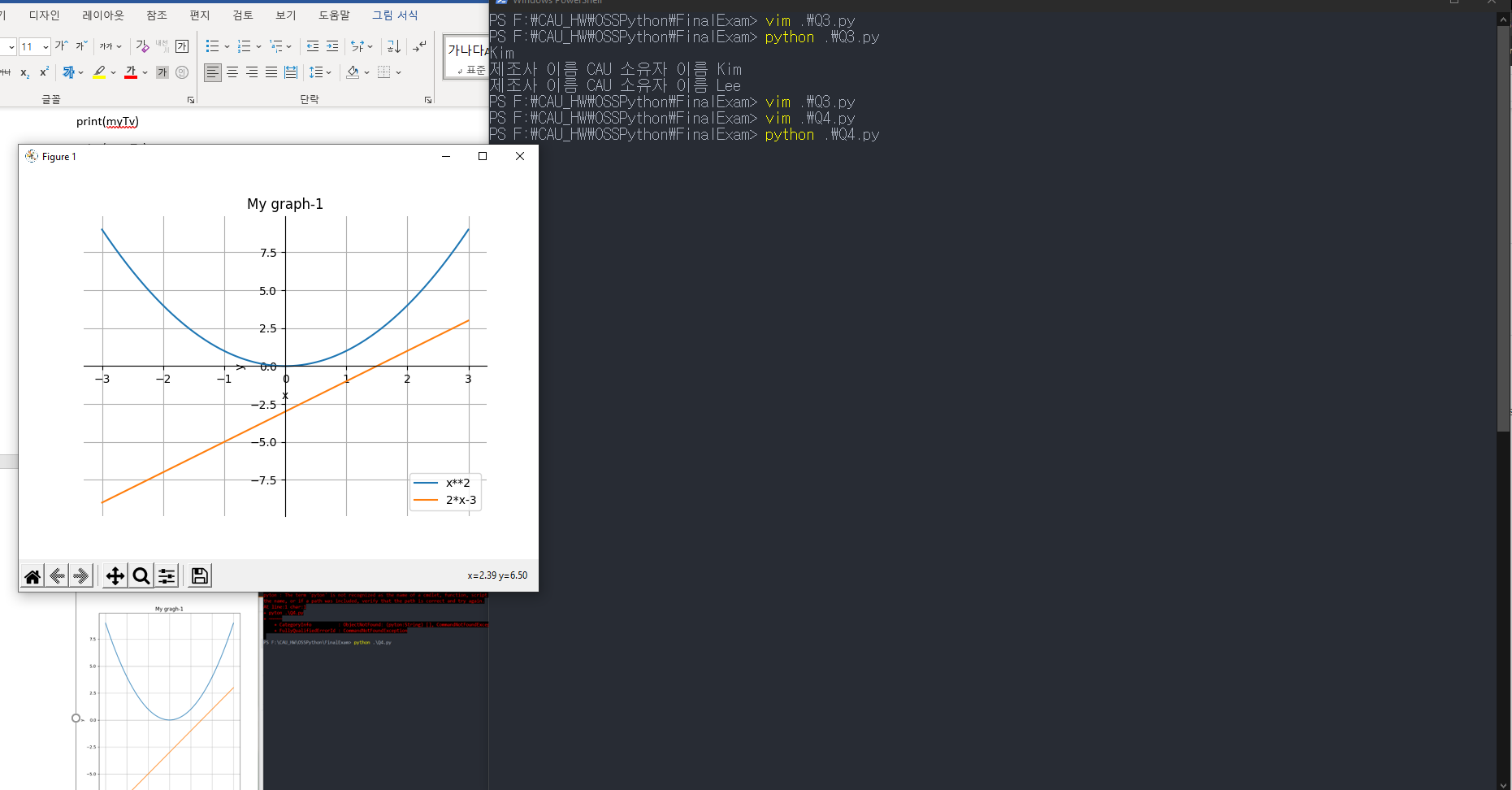
myTv.setOwner("Kim")

yourTv.setOwner("Lee")

print(myTv.getOwner())

print(myTv)

print(yourTv)

[Question-4 (5 points)]  
  
import matplotlib.pyplot as plt

import numpy as np

X=np.arange(-3,3,0.001)

Y1=[x\*\*2 for x in X]

Y2=[2\*x-3 for x in X]

ax=plt.axes()

ax.plot(X,Y1,label="x\*\*2")

ax.plot(X,Y2,label="2\*x-3")

ax.set\_title("My graph-1")

ax.spines.left.set\_position('center')

ax.spines.right.set\_color('none')

ax.spines.bottom.set\_position('center')

ax.spines.top.set\_color('none')

plt.legend()

plt.grid()

ax.set\_xlabel('x')

ax.set\_ylabel('y')

plt.show()

[Question-5 (2points)]  
Radio Button의 경우 보기의 리스트 중 하나의 옵션만 선택할 수 있는 위젯이다. 만약 다른 옵션이 선택될 경우 원래 선택되었던 옵션이 꺼져 전체 보기 중 단 하나의 보기만 선택되도록 고안된 버튼이다. 따라서 Radio Button은 유저가 하나의 보기만 골라야 할 때 사용해야 한다.

Check Box의 경우 보기의 리스트 중 중복하여 보기를 선택할 수 있는 위젯이다. 만약 다른 옵션이 선택될 경우 원래 선택되었던 옵션은 유지되며 다수의 옵션이 선택되거나 해제될 수 있다. 이를 통해 Check Box를 사용하면 유저에게 다양한 선택의 조합을 받아오고 싶을 때 사용한다는 것을 알 수 있다.

[Question-6 (3points)]

Array는 정적 할당에 해당하며, List의 경우는 동적 할당에 해당한다. 그렇기 때문에 List는 마음대로 크기를 조절할 수 있으며 데이터 추가 삭제가 자유롭다. 하지만 Array는 선언할 때 부터 크기를 가지고 선언하기 때문에 List처럼 자유로운 수정이 불가능하다. 또한 Array는 메모리 주소가 붙어있는 반면 List는 떨어져 있을 수 있는 형태이다. 또한 Array의 경우 numpy에 미리 저장된 C코드를 기반으로 연산되어 속도가 더 빠르며 벡터화되어 계산되는 형식이다.

[Question-7 (2points)]  
shape : (1,1,1)  
ndim : 3

[Question-8 (3points)]  
Try : 실행하고자 하는 코드를 넣고 이 코드에 에러가 발생하는 지 실행해보면서 테스트를 할 수 있다.  
Except : Try에서 실행한 코드에서 에러가 발생한 경우 실행하고 싶은 코드를 입력하여 예외가 발생할 경우 대응응할 수 있다.  
Else : Try 절에서 모든 코드가 정상적으로 진행되었다면 Except의 어떠한 코드도 실행되지 않고 Else의 코드가 실행된다.  
Finally : Try 절에서 에러가 발생하는 것과 상관 없이 항상 실행되는 코드를 넣어 시도가 끝나는 것을 알려줄 수 있다.  
  
[Question-9 (3points) ]  


import time

from datetime import date

day=["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday","Sunday"]

secs = time.time()

tm = time.localtime(secs)

print("Today is " + str(tm.tm\_year) + "/" +str(tm.tm\_mon)+"/"+str(tm.tm\_mday)+", "+str(day[date(tm.tm\_year, tm.tm\_mon, tm.tm\_mday).weekday()]))