2018-6-15

Lab Report 1

Saltzman,Thomas Renner; Fei Wu; Widner,John Drury

1. Introduction

In the first two week lab assignment 1, 6 program questions were designed to require us to get familiar with python basic topics: loop, functions, list, class and the NumPy from Python package.

1. Objectives

Lab Objects:

1. For any web application login, the user password need to be validated against database rules. For My UMKC web application following are the criteria for valid password:

a. The password length should be in range 6-16 characters

b. Should have at least one number

c. Should have at least one special character in [$@!\*]

d. Should have at least one lowercase and at least one uppercase character

Use loops to write a python program for the above scenario. No need to create web application, only validation is sufficient.

2. Write a Python function that accepts a sentence of words from user and display the following:

a. Middle word

b. Longest word in the sentence

c. Reverse all the words in sentence

3. Given a list of n number, write a Python program to find triplets in the list which gives the sum of zero.

Sample input: [1, 3, 6, 2, -1, 2, 8, -2, 9]

Sample output: [(3, -1, -2)]

4. Consider the following scenario. You have a list of students who are attending class "Python" and another list of students who are attending class "Web Application".

Find the list of students who are attending both the classes. Also find the list of students who are not common in both the classes. Print the both lists.

5. Write a python program to create any one of the following management systems. You can also pick one of your own.

a. Library Management System (e.g. classes Person, Student, Librarian, Book etc.)

b. Airline Booking Reservation System (e.g. classes Flight, Person, Employee, Passenger etc.)

c. Hotel Reservation System (e.g. classes Room, Occupants, Employee etc.)

d. Student Enrollment System (e.g. classes Student, System, Grades etc.)

e. Expense Tracker System (e.g. classes Expense, Transaction Category etc.)

Prerequisites:

a. Your code should have at least five classes

b. Your code should have \_init\_ constructor in all the classes

c. Your code should show inheritance at least once

d. Your code should have one super call

e. Use of self is required

f. Use at least one private data member in your code

g. Use multiple Inheritance at least once

h. Create instances of all classes and show the relationship between them

6. Using NumPy create random vector of size 15 having only Integers in the range 0-20. Write a program to find the most frequent item/value in the vector list.

1. Approaches/Methods

In question 1, we input the password (str(input()), then we use a while loop to test all requirements are met(exit if not).First we test the length(len()) to ensure it is range 6-16. Then we use method (seach.()) to check if the password contain upper/lower letters, special case(!@#$) and numbers to validate the input password.

In question 2, we split the input sentence and then convert it to a list using .split(), the sentence should be spate by comma, a method of len() to get the length of this list and then sort the list based on the length(sort()), if the length is even (len()%2==0), print the two middle indices(int(length/2-1)], myList[int(length/2)], or print the middle index. Since the list was sorted, we can easily get the longest word in the list(max()). Create an empty string and add the old one with reversed order to the new string (word[::-1].lower()).

In question 3, again check the len() of the input list, then use three if loops and check one by one, if sum of three elements equals zero then print the three numbers(from index of the list).

In question 4, We create a list of persons with random names using .append, randomly place a student in non one or both class list using a for loop, then print the students name and its memory location", notice: we have turned all list into set and put the students in both classes in one set, whereas students who did register classes in another set.

In question 5, we create a 7 classes(flight, person, employee, passenger, itinerary, payment and airticket) and each class has \_init\_ constructor and self. , where class of employee and passenger inherit the properties from class of person. Class of Airticket relates to class(Itinerary, Payment) as an example of multiple Inheritance, in class of person, the birthday (private data) can not be accessed by other classes using \_ \_name. Instances were shown as in the main() that uses all the class information.

In question 6, we generate a random array from 0 to 20 of size 15 using the method np.random.randint(). And then create an array from 0 to max value in the array and counts the frequency of that value and stores the count at an index of that value by applying np.bincount().

1. Workflow

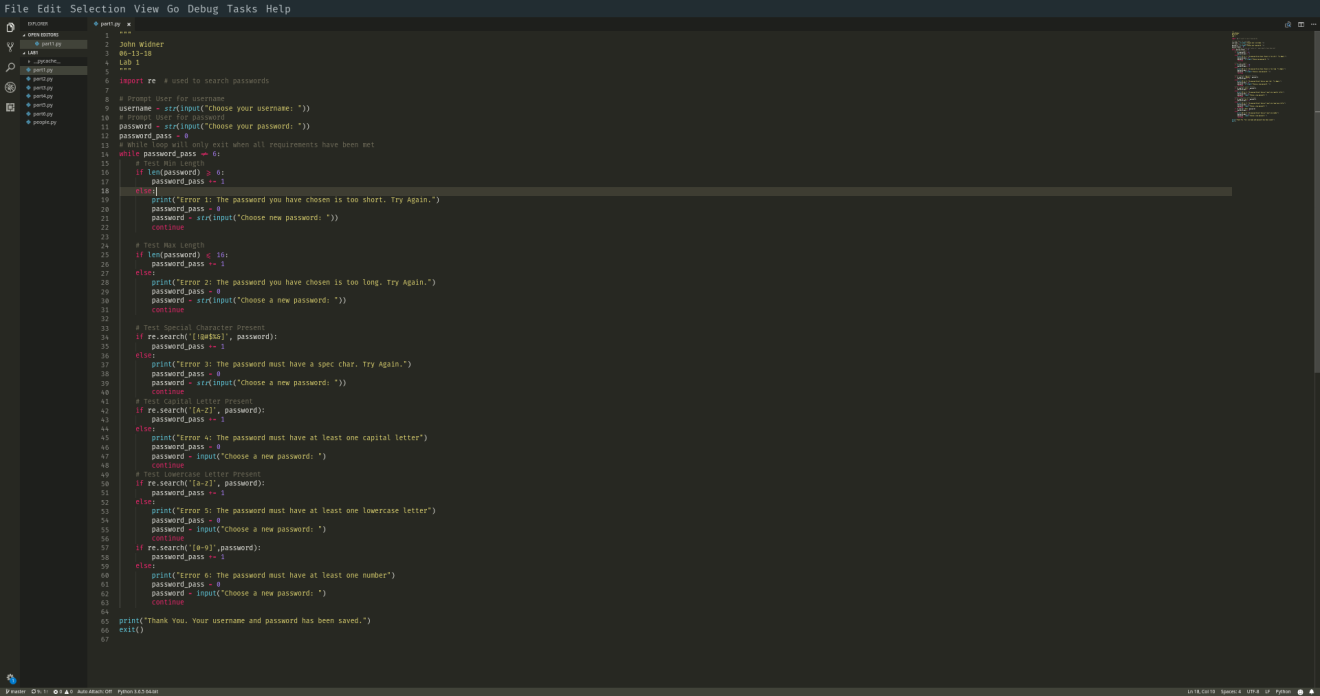
Until now the four python classes covered all topics with respect to the lab 1 assignment. After each class, we reviewed the lecture and started with the related question in the lab. For each question, we worked independently and try to solve the problem, we created our own ideas, coded, discussed and finally picked a best answer, and everyone was participating in this lab with hundred percentage effort.

1. Datasets (if applicable)

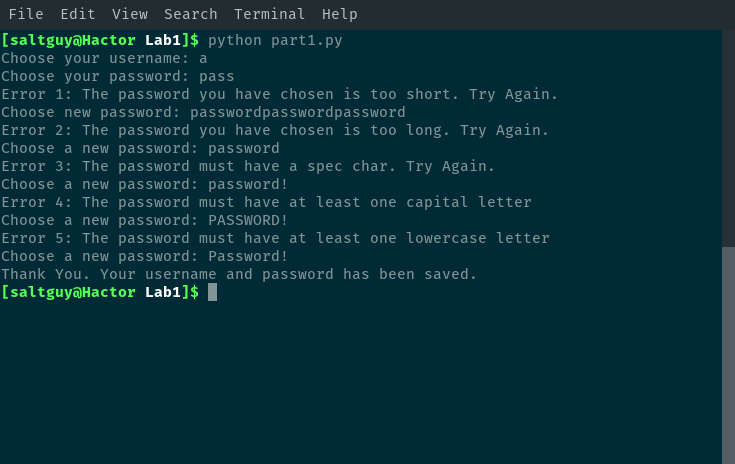
NA.

1. Parameters/Results

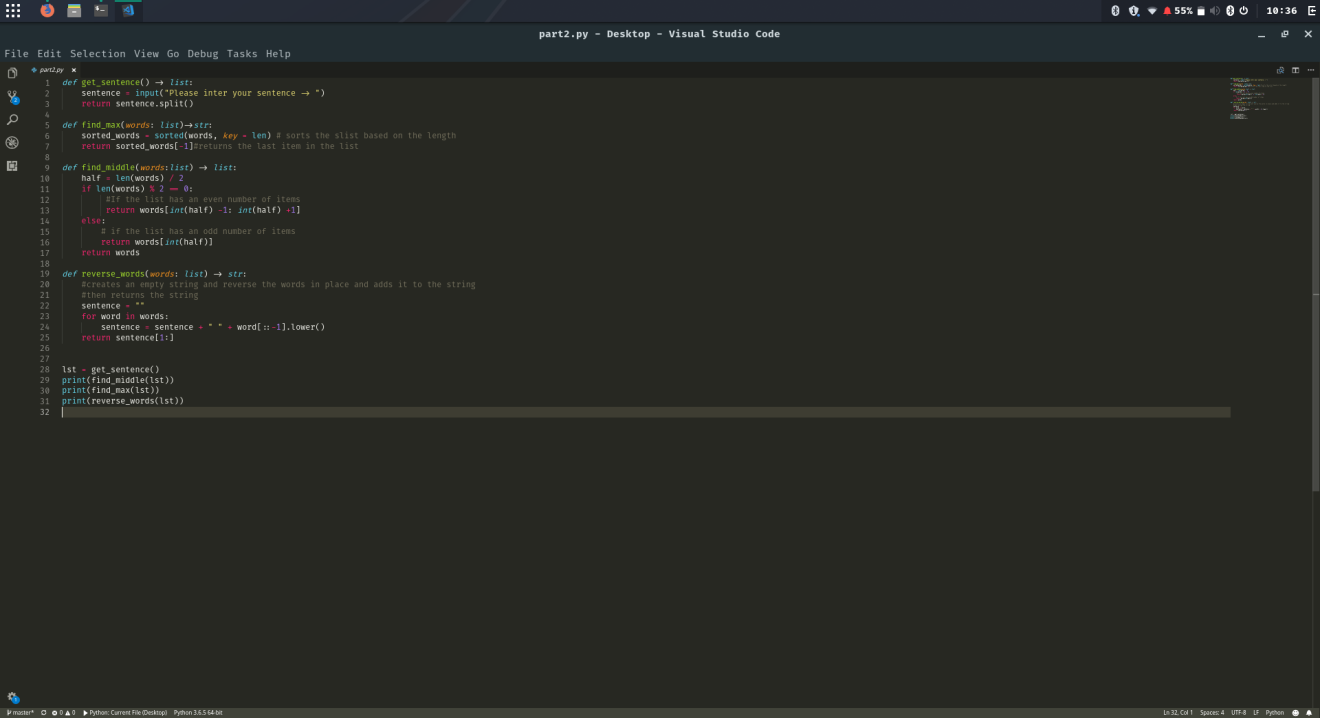
Part1:code



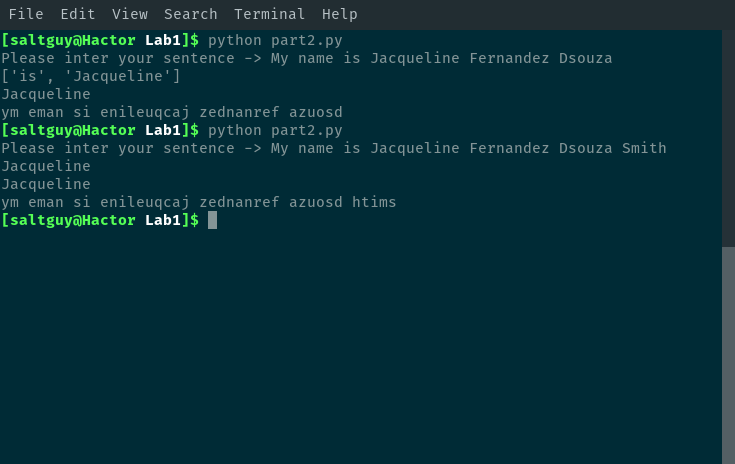
Part1: output



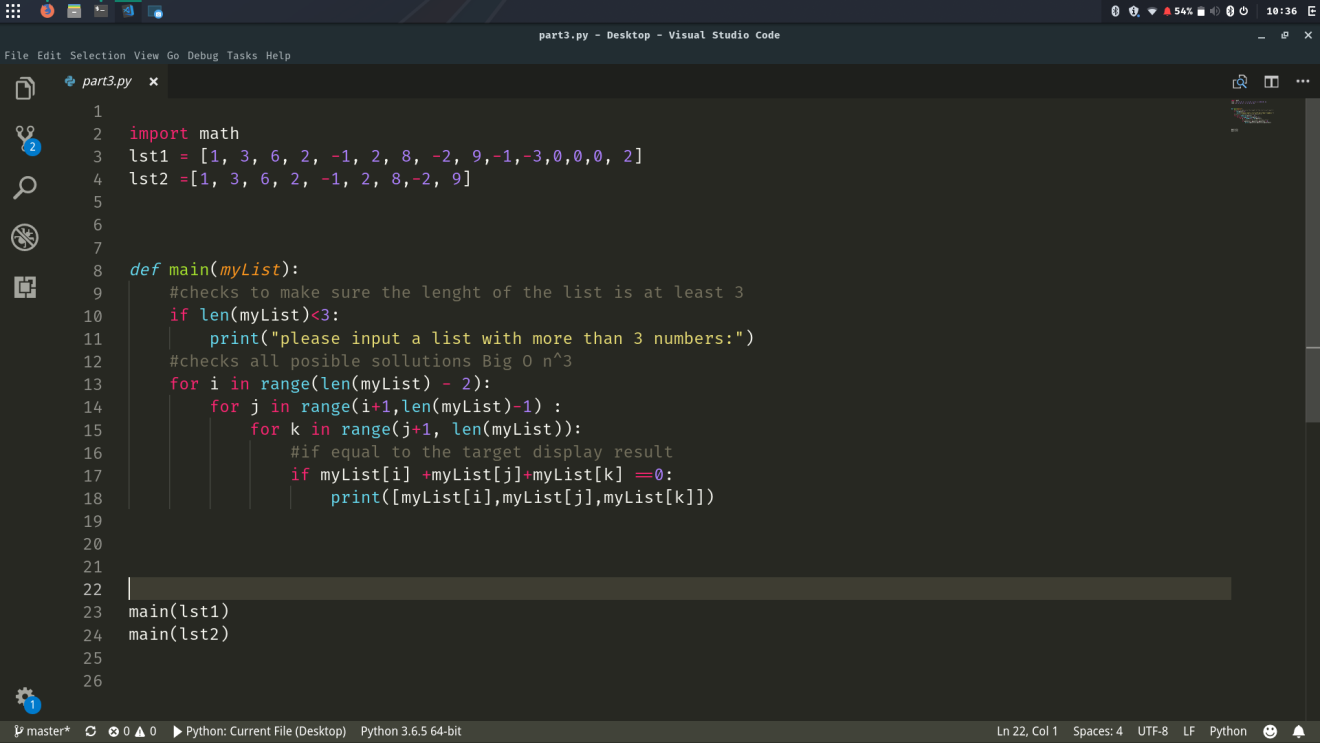
Part2:code



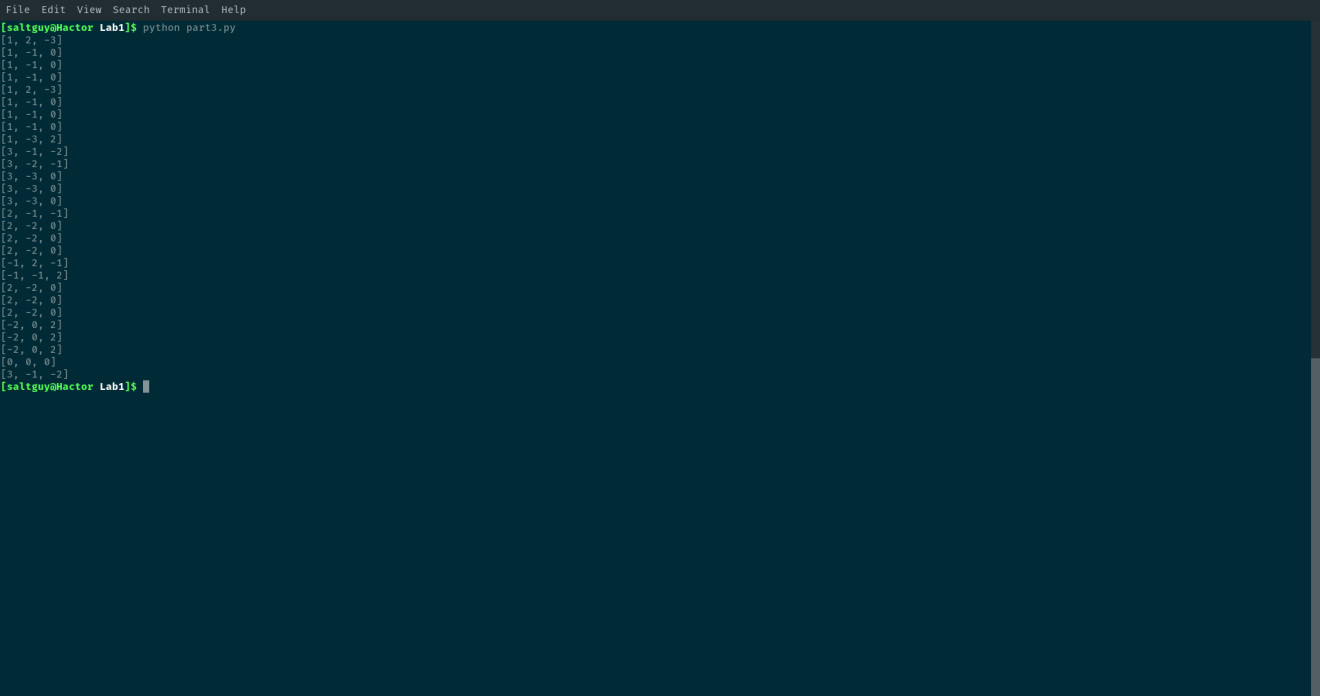
Part2: output



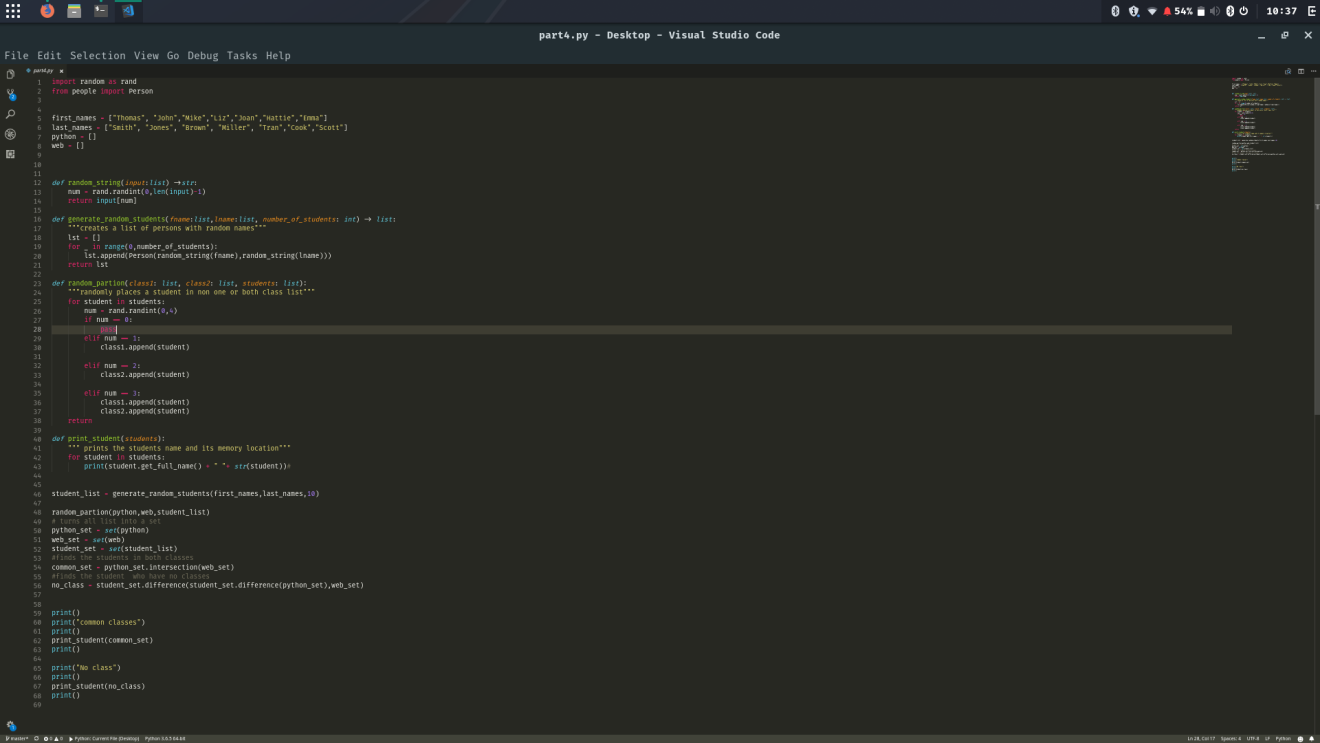
Part3:code



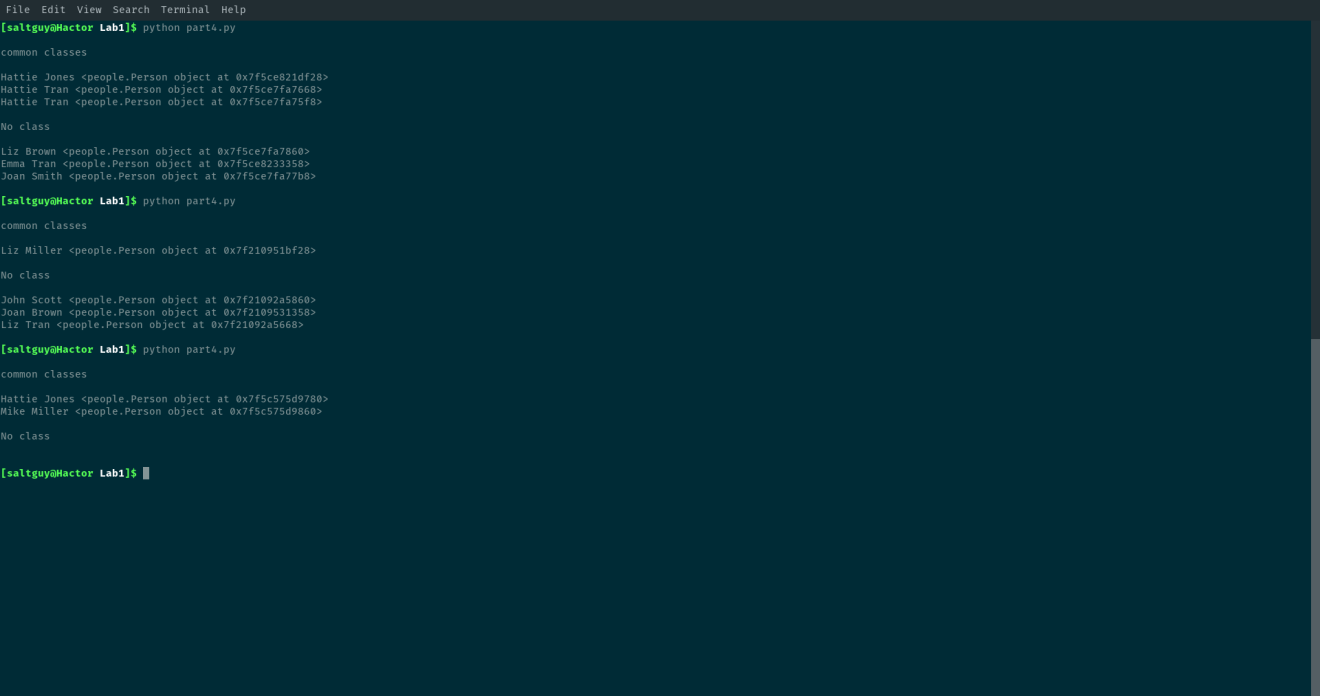
Part3: output

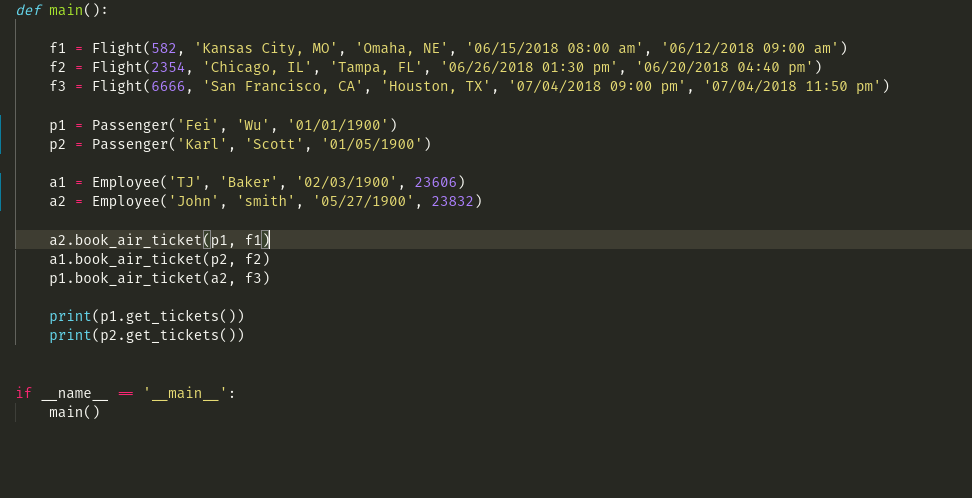


Part4:code

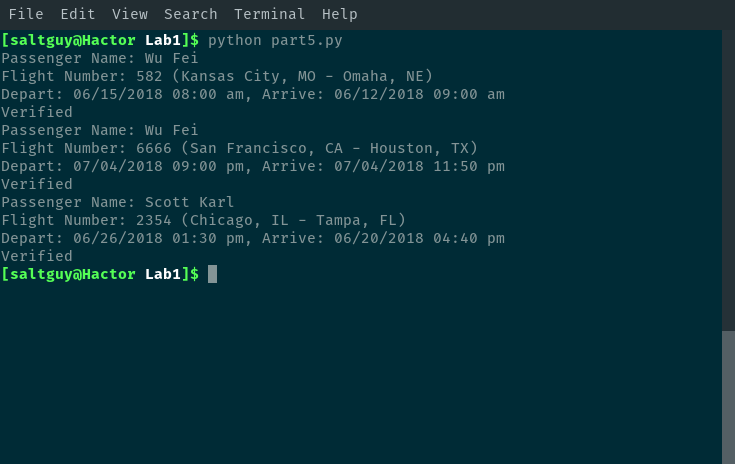


Part4: output

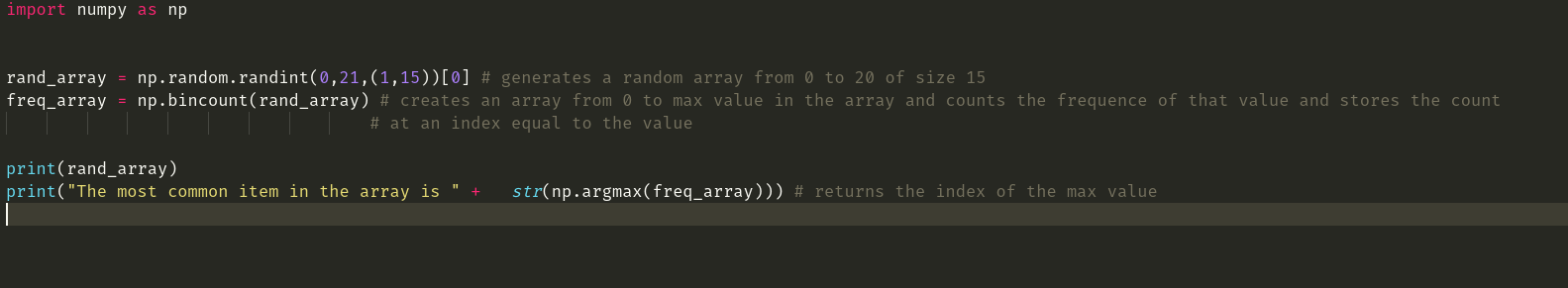


Part5:code

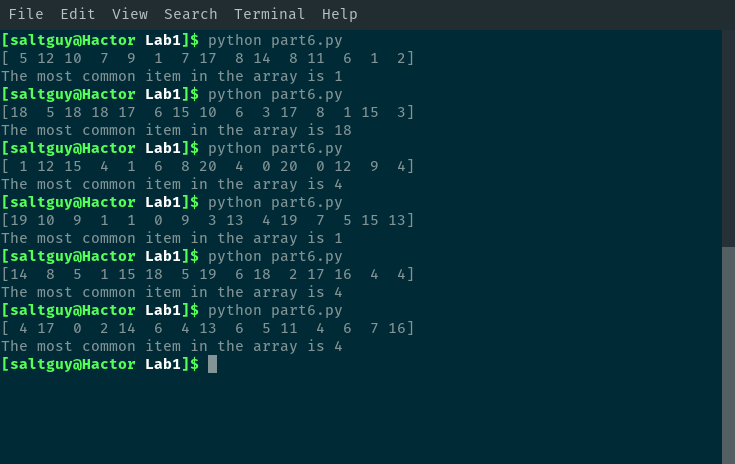
Part5: output



Part6:code



Part6: output



1. Evaluation & Discussion

We have evaluated our code and results from various angles: First and most import: make sure all code is runnable and produces correct results. Second, try several different inputs to insure that all edge cases are covered, for instance, in question 3, there may be more than one output from the list that the sum of triplet equals to zero. Third, how to improve the code is another challenge in this lab, our goal is to focus on effective, concise, and readable code. We compare our work with other similar (not same) examples online and extract their best parts, learn the principle behind their ideas and apply it to our work. Four, could we design a better algorithm to solve the problem to reduce the time/space complexity? For example, in question 3, our complexity is O(n3) but is a more simple algorithm. If we use hashing or sorting method, the complexity can be decreased to O(n2) but needs more complex algorithm.

1. Conclusion

All the requirements in lab 1 were met, and more program skills are needed which can be obtained by more practices.