PROBLEM 1 : (***What is the output? (16 pts) (6 minutes)***)

For the following code, write the output to the right of each print statement.

```
                                                    OUTPUT:
lista = ['la', 'so', 'be', 'if']
print(sorted(lista))
#-------------------------------------------------

lista = ['wok','table','dish','rug']
listb = [(w, len(w)) for w in set(lista)]
print(sorted(listb))
#-------------------------------------------------

lista = [14, 6, 21, 17]
listb = sorted(lista, reverse=True)
print(listb)
#-------------------------------------------------

lista = ['green', 'gray', 'red', 'pink', 'blue']
listb = sorted(lista, key=len)
print(listb)
#-------------------------------------------------

lista = [(7,8,5),(3,9),(10,5,1)]
listb = sorted(lista, key=min)
print(listb)
#-------------------------------------------------

lista = [(11,4,7),(9,10,5),(6,1,8)]
listb = sorted(lista, key=lambda x:x[2])
print(listb)
#-------------------------------------------------

d = {'T': [9,3], 'G':[6,0], 'B':[7,5]}
ans = sorted(d.keys())
print(ans)
ans = sorted(d.items(), key=lambda x:x[1][0])
print(ans[0])
```

PROBLEM 2 : (***Short problems (18 pts) (14 minutes)***)

Answer questions about or complete the following functions. This problem has three parts. Your functions should work for any valid data, not just the examples shown.

PART A (6 pts) (4 minutes)

Assume d is a dictionary in which each key is a string, and each value is a list of three integers.

Consider the following function named `mysteryD2` that has two parameters, a dictionary named `dict`, where each key is a string mapped to a list of three integers, and a parameter named `num` that is an integer.

```
1 def mysteryD2(dict, num):
2     alist = []
3     for key in dict:
4         alist.append(dict[key][num])
5     return alist
```

QUESTION 1 Consider that dict is the dictionary shown below.

```
dict = {"A": [5, 3, 1], "M": [6, 9, 7], "Q": [2, 8, 4]}
```

What does mysteryD2(dict, 2) return?

QUESTION 2

Explain in words what the function mysteryD2 does? (Briefly, in 1-2 sentences)

QUESTION 3

For the code above, show below how dict.items() can be used instead in line 3, and how line 4 then can be shortened, **change both lines 3 and 4**, in the space provided in the code below (more than two lines of space is provided to give you space to write.) With these changes, the result should be the same as the code above.

```
def mysteryD2(dict, num):
    alist = []




    return alist
```

PART B (6 pts) (4 minutes)

Consider the following function named `mystery2` that has two parameters, a string named `word` representing a word, and a parameter named `wordlist` that is a list of strings, where each string is one word.

```
1 def mystery2(word, wordlist):
2     answer = []
3     for someword in wordlist:
4         diff = len(someword) - len(word)
5         answer.append((someword, diff))
6     return answer
```

QUESTION 1 Consider that word and wordlist are defined below.

```
word = "heart"
wordlist = ['red','announcement','slurp','golf','strike']
```

For this input, what does mystery2(word, wordlist) return?

QUESTION 2

Explain in words what the `mystery2` function does. (Briefly, in 1-2 sentences)

QUESTION 3

Suppose we want the return value sorted in reverse order by the numbers. Explain which line of code you would modify and give that code.

PART C (6 pts) (6 minutes)

Write the function named `ordering2` that has one parameter named `lista` that is a list of tuples, where each tuple has three items: first an integer, second a string, and third an integer.

This function returns a list of the tuples sorted in the following way:

1. sorted by the middle item in each tuple, the string, in reverse alphabetical order
2. break ties by sorting the last item in each tuple, an integer, in reverse order
3. after 1) and 2) break ties sorting by the first item in each tuple, an integer

```
For example, the call ordering2([(8,'c',2),(8,'k',2),(5,'s',9),(6,'c',2),(5,'s',7)])
returns the list:
[(5,'s',9),(5,'s',7),(8,'k',2),(6,'c',2),(8,'c',2)]
```

Complete the function below.

```
def ordering2(lista):
```

PROBLEM 3 : (***Favorite Restaurants (48 pts) (48 minutes)***)

This problem is about data related to favorite restaurants.

There are six functions to write in this part. Your functions should work for any valid data, not just the examples shown.

Most of the problems have `datalist` as one of the parameters. The parameter datalist is a list of lists, with each inner list representing information about one student and three or more of their favorite restaurants in order by most favorite first. More specifically, each inner list has

1. a string representing a student name,
2. an integer representing the student's id number
3. a string representing the name of the student's dorm
4. a list of at least three strings of the student's favorite restaurants in order by most favorite first

For example, assume datalist is the lists of lists shown below. The first inner list represents the student named Johnson, whose student id number is 345128, whose dorm is Southgate, and who has three favorite restaurants: their favorite restaurant is The Commons, their second favorite restaurant is McDonalds, and their third favorite restaurant is Gothic Grill.

Note that each student name is unique and appears only once in datalist. Also note that each student has at least three favorite restaurants, some have more than three.

```
datalist = [
 ['Johnson', 345128, 'Southgate', ['The Commons', 'McDonalds', 'Gothic Grill']],
 ['Borden', 565321, 'Keohane', ['Sprout', 'Il Forno', 'The Commons', 'Guglhupf']],
 ['Krishna', 476284, 'Alspaugh', ['Sushi Love', 'Tandoor', 'Farmstead', 'Sprout']],
 ['Zhuong', 237645, 'Giles', ['Tandoor', 'Sprout', 'Sushi Love', 'Gothic Grill']],
 ['Kerola', 426712, 'Edens', ['Tandoor', 'The Commons', 'Il Forno']],
 ['Proctor', 234712, 'Keohane', ['Zwelis', 'Sprout', 'Tandoor', 'Il Forno']],
 ['Scheyer', 346353, 'Alspaugh', ['Zwelis','The Commons','Sushi Love','Gothic Grill']],
 ['Williams', 346372, 'Giles', ['Gothic Grill', 'Sprout', 'Zwelis']],
 ['Hill',145672,'Keohane', ['Sushi Love','Tandoor','The Commons','Sprout','Il Forno']]
 ]
```

In solving the problems that follow, you may call any of the other functions in this problem.

Go to the next page to start Part A of this problem.

Part A (8 pts) (8 minutes)

Write the function named **peopleWhoLikeRestaurant** that has two parameters, one named **datalist**, which is a list of lists in the format described earlier, and a string named **restaurant**, which is the name of a restaurant.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function calculates a unique sorted list of students for which the parameter restaurant is one of their favorite restaurants.

For example, assume datalist is the list of lists shown on the first page of this problem. The call `peopleWhoLikeRestaurant(datalist,"The Commons")` returns the list: `['Borden', 'Hill', 'Johnson', 'Kerola', 'Scheyer']`.

Complete the function below.

```
def peopleWhoLikeRestaurant(datalist, restaurant):
```

Part B (8 pts) (8 minutes)

Write the function named **favRestaurantsFromDorm** that has two parameters. The first parameter is named **datalist**, which is a list of lists in the format described earlier, and the second parameter named **dorm** is the name of a dorm.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function returns a **sorted unique** list of favorite restaurants that are in datalist and for only students who are in the parameter dorm specified. Using the datalist shown on the first page of this problem, if the dorm is 'Giles' then the call `favRestaurantsFromDorm(datalist, 'Giles')` would return the list: `['Gothic Grill', 'Sprout', 'Sushi Love', 'Tandoor', 'Zwelis']`

Complete the function below.

```
def favRestaurantsFromDorm(datalist, dorm):
```

Part C (8 pts) (8 minutes)

Write the function named **countRestaurantsPeopleLike** that has one parameter named **datalist**, which is a list of lists in the format described earlier.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function returns a sorted list of tuples, where each tuple has two items. The first item is a string that is the name of a student, and the second item is an integer that is the number of favorite restaurants the student has. The tuples are sorted first by the second item, the count, in decreasing order. Ties are broken by the first item in alphabetical order.

For example, using the datalist described on the first page of this problem, the call `countRestaurantsPeopleLike(datalist)` returns the list of tuples: `[('Hill', 5), ('Borden', 4), ('Krishna', 4), ('Proctor', 4), ('Scheyer', 4), ('Zhuong', 4), ('Johnson', 3), ('Kerola', 3), ('Williams', 3)]`.

```
def countRestaurantsPeopleLike(datalist):
```

Part D (8 pts) (8 minutes)

Write the function named **topRestaurantMoreThanOnce** that has one parameter named **datalist**, which is a list of lists in the format described earlier.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function returns a **sorted unique list** of restaurants that appear as the favorite restaurant for **at least two people**. Favorite restaurant means it appears as the first restaurant in a student's list of favorite restaurant.

For example, using the datalist on the first page of this problem, the call `topRestaurantMoreThanOnce(datalist)` returns sorted list: `['Sushi Love', 'Tandoor', 'Zwelis']`.

Complete the function below.

```
def topRestaurantMoreThanOnce(datalist):
```

Part E (8 pts) (8 minutes)

Write the function named **dictRestaurantToListRankings** that has one parameter named **datalist**, which is a list of lists in the format described earlier.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function returns a dictionary mapping the name of a restaurant to a list of its integer rankings (1 if it was listed first by a student, 2 if it was listed second by a student, etc.) from datalist.

For example, using the datalist on the first page of this problem, the restaurant 'Sushi Love' was rated as the top restaurant by two students, and as the third favorite restaurant by two students, so its rankings would be [1, 3, 3, 1] (the order of the numbers does not matter). The call `dictRestaurantToListRankings(datalist)` returns the dictionary:

```
{'The Commons': [1, 3, 2, 2, 3], 'McDonalds': [2], 'Gothic Grill': [3, 4, 4, 1],
'Sprout': [1, 4, 2, 2, 2, 4], 'Il Forno': [2, 3, 4, 5], 'Guglhupf': [4],
'Sushi Love': [1, 3, 3, 1], 'Tandoor': [2, 1, 1, 3, 2], 'Farmstead': [3],
'Zwelis': [1, 1, 3]}
```

```
def dictRestaurantToListRankings(datalist):
```

Part F (8 pts) (8 minutes)

Write the function named **restaurantWithTopScore** that has one parameter named **datalist**, which is a list of lists in the format described earlier.

We repeat the format of parameter datalist, which is a list of lists. Each inner list has 1) a string representing a student name, 2) an integer representing the student's id number 3) a string representing the name of the student's dorm and 4) a list of at least three strings of the student's favorite restaurants in order by most favorite first.

This function returns the name of the restaurant with the top score. If there is a tie, return any of the restaurants that received the top score. A restaurant's score is calculated by giving it 5 points for being the first restaurant in a student's favorite list, 3 points for being the second favorite restaurant in a student's list and 1 point for being the third favorite restaurant in a student's list. Restaurants listed as 4th favorite or beyond receive no points. For example if a restaurants ranking list was [2, 4, 3, 1, 3] then this restaurant's score would be 5 points for the 1 ranking, 3 points for the 2 ranking, 1 pt for each of the 3 rankings and no points for the 4 ranking, so 5 + 3 + 1 + 1 or a score of 10.

For example, consider the datalist example given at the beginning of this problem. The call `restaurantWithTopScore(datalist)` returns Tandoor as the restaurant with the highest score. Its' score is 17.

You may call any of the other functions in this problem to solve this part.

```
def restaurantWithTopScore(datalist):
```