Name____StudentID____________________

Level 1 [1.25 Point]: (Save program under filename expense1.py)

Level 1.1 [0.25 Point] (Save program under filename expense1.py

Write a program that compares two numbers and reports whether the first number is less than, more than, or equal to the second number.

Function Name	Inputs	Outputs	Functionality
compare_nums	two numbers	string	Receive two numbers as input parameters. Compare two numbers, and return “less than” if the first number is less than the second number. Or return “more than” if the first number is more than the second number. Otherwise, return “equal to”.

Sample outputs are shown below:

Level 1,1, Sample output 1:	**Level 1.1, Sample output 2:**	**Level 1.1, Sample output 3:**
Enter x: 2 Enter y: 3.5 2.0 is less than 3.5	Enter x: 7.2 Enter y: 4 7.2 is more than 4.0	Enter x: 6 Enter y: 6 6.0 is equal to 6.0

Level 1.2 [1.00 Point] (Continue to work on your program under filename expense1.py)

Each day, teachers will spend money on stationery (เครื่องเขียน).

Write a program to compare stationery expense (ค่าใช้จ่าย) of two teachers.

First, the program reads number of days to compare the expenses. Then, the program reads expenses of n days for Teacher 1 and Teacher 2 respectively. At the end, the program will report whether Teacher 1 spends more, less, or equal to money of Teacher 2 on every day.

In your program, you must include the following 3 functions:

* Your program can have more than these 3 functions.

Function Name	**Inputs**	**Outputs**	**Functionality**
compare_nums ***(same as level 1.1)***	two numbers	string	Receive two numbers as input parameters. Compare them, and return “less than” if the first number is less than the second number. Or return “more than” if the first number is more than the second number. Otherwise, return “equal to”.
read_list	One number (n = number of days)	A list of numbers	Receive n (number of days) as input parameter. Read n expenses of one teacher from user, and return a list of n expenses.

print_list_comparison	two lists of numbers	None	Receive two lists of expenses as input parameters. Compare and report whether Teacher 1 spends more, less or equal to Teacher 2 on every day.

Sample outputs are shown below.

Level 1.2, Sample output 1:	**Level 1.2, Sample output 2:**
Enter #days: 3 Teacher 1: Day 1: enter expense: 50 Day 2: enter expense: 20 Day 3: enter expense: 40 Teacher 2: Day 1: enter expense: 30 Day 2: enter expense: 25 Day 3: enter expense: 15.5 [50.0, 20.0, 40.0] [30.0, 25.0, 15.5] Day 1: Teacher1 spends more than Teacher2 Day 2: Teacher1 spends less than Teacher2 Day 3: Teacher1 spends more than Teacher2	Enter #days: 4 Teacher 1: Day 1: enter expense: 40 Day 2: enter expense: 40 Day 3: enter expense: 50 Day 4: enter expense: 10 Teacher 2: Day 1: enter expense: 25 Day 2: enter expense: 40 Day 3: enter expense: 80 Day 4: enter expense: 25 [40.0, 40.0, 50.0, 10.0] [25.0, 40.0, 80.0, 25.0] Day 1: Teacher1 spends more than Teacher2 Day 2: Teacher1 spends equal to Teacher2 Day 3: Teacher1 spends less than Teacher2 Day 4: Teacher1 spends less than Teacher2

Level 2 [2.5 Points]: (Save program under filename expense2.py. This Level 2 can be done without completing Level 1)

Write a program to compute summation of expenses from some specific days, *not every day*.

First, the program shows expenses of two teachers. Then, the program continuously reads which days user wants to compute summation until user enters -99. Next, the program reads which teacher to compute summation, and reports summation of some specific days with 2 decimal numbers. The program asks whether the user wants to continue computing the next summation or not. If user enter ‘n’, the program will stop.

Note that beginning of main part for Level 2 is given (see code).

In your program, you must include the following 2 functions:

* Your program can have more than these 2 functions.

Function Name	Inputs	Outputs	Functionality
read_day_list	None	A list of days	Read which days that user wants to compute summation. Return a list of days.
compute_partial_sum	A list of days	One number	Receive a list of days as input parameter. Compute and return summation of expenses from a list of days.

Beginning of main for each sample output are given below.

Level 2, Sample output 1:
Beginning of main

```python
# main
t1 = [50, 20, 30]
t2 = [40, 25, 15.5]
print('Teacher 1:')
print(t1)
print('Teacher 2:')
print(t2)

# fill in the rest of program
yourself
```

Level 2, Sample output 2:
Beginning of main

```python
# main
t1 = [40,40,50,15,12.5]
t2 = [25,40,80,20,10]
print('Teacher 1:')
print(t1)
print('Teacher 2:')
print(t2)

# fill in the rest of program
yourself
```

Sample outputs are shown below.

Level 2, Sample output 1:

```
Teacher 1:
[50, 20, 30]
Teacher 2:
[40, 25, 15.5]
Compute sum of some days...
Enter day number: 1
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 2
[1, 3]
[40, 25, 15.5]
Sum of Days [1, 3] = 55.50
Continue (y/n): y

Compute sum of some days...
Enter day number: 2
Enter day number: -99
Which teacher (1 or 2): 1
[2]
[50, 20, 30]
Sum of Days [2] = 20.00
Continue (y/n): y

Compute sum of some days...
Enter day number: 2
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 2
[2, 3]
[40, 25, 15.5]
Sum of Days [2, 3] = 40.50
Continue (y/n): n
```

Level 2, , Sample output 2:

```
Teacher 1:
[40, 40, 50, 15, 12.5]
Teacher 2:
[25, 40, 80, 20, 10]
Compute sum of some days...
Enter day number: 1
Enter day number: 3
Enter day number: 5
Enter day number: -99
Which teacher (1 or 2): 1
[1, 3, 5]
[40, 40, 50, 15, 12.5]
Sum of Days [1, 3, 5] = 102.50
Continue (y/n): y

Compute sum of some days...
Enter day number: 1
Enter day number: 2
Enter day number: 4
Enter day number: 5
Enter day number: -99
Which teacher (1 or 2): 2
[1, 2, 4, 5]
[25, 40, 80, 20, 10]
Sum of Days [1, 2, 4, 5] = 95.00
Continue (y/n): n
```

Level 3 [1.75 points] (Save program in another file from Levels 1,2, filename = expense3.py)

In this Level 3, expense files for two teachers are given. (If you do not know how to read file, you can use read_list function and enter expenses yourself.)

There are 3 data sets: sets A, B, C. Each data set contains 2 files for Teacher1 and Teacher2 respectively. When we run the program, only one data set (or 2 files) is used. The data sets are shown below:

teacher1A.txt	**teacher2A.txt**
50	40
20	25
30	15.5
teacher1B.txt	**teacher2B.txt**
40	25
40	40
50	80
15	20
12.5	10
teacher1C.txt	**teacher2C.txt**
100	80
52	124
84	67
10.5	20.75
20.25	5.5
13	33
157	120
40	71
68	68
76	46

Combine your code from Levels 1 and 2 into Level 3 here. Add the code to report which day the teacher spends the maximum expense, and the maximum amount. See sample outputs for more understanding.

The program starts by asking user which data set he/she wants to run (set A,B or C). Then, the program prints menu choices 1-3 and 0.

If user chooses 1, the program will compare how Teacher1 spends more, less, or equal to Teacher2.

If user chooses 2, the program will compute summation of expenses on specific days.

If user chooses 3, the program will report which day the specific teacher spends the maximum expense and the maximum amount.

After reporting result for choices 1,2 or 3, the program will continuously ask for the next menu choice from user. If user chooses 0, the program will exit.

Note that beginning of main part for Level 3 is given (see code).

In your program, you must include the following 3 functions:

* Your program can have more than these 3 functions.

Function Name	Inputs	Outputs	Functionality
read_file	string (filename)	A list of expenses	Receive filename as input parameter. Read expenses of one teacher from a file and return a list of expenses.
read_menu_choice	None	Menu choice	Print menu choices. Read choice of menu from user and return this choice.
find_max	A list of expenses	Day that teacher spends the maximum expense, and the maximum amount	Find which day that the specific teacher spends maximum expense. Return such day and the maximum amount.

Beginning of main is given below.

Beginning of Main

```
#main

data_set = input('Enter your data set (A,B,C): ')
filename1 = 'teacher1'+data_set+'.txt'
filename2 = 'teacher2'+data_set+'.txt'
t1 = read_file(filename1)
t2 = read_file(filename2)

#fill in the rest of program yourself
```

Sample outputs are shown below: *Note that sample outputs only show results for data sets A and C. However, your code should work with data set B also.*

Level 3, Sample output 1:	**Level 3, Sample output 2:**
Enter your data set (A,B,C): A Teacher 1: [50.0, 20.0, 30.0] Teacher 2: [40.0, 25.0, 15.5] Menu: 1=List comparison 2=Partial sum 3=Find max 0=Quit Enter your choice: 1 Day 1: Teacher1 spends more than Teacher2 Day 2: Teacher1 spends less than Teacher2 Day 3: Teacher1 spends more than Teacher2 Menu: 1=List comparison 2=Partial sum	Enter your data set (A,B,C): C Teacher 1: [100.0, 52.0, 84.0, 10.5, 20.25, 13.0, 157.0, 40.0, 68.0, 76.0] Teacher 2: [80.0, 124.0, 67.0, 20.75, 5.5, 33.0, 120.0, 71.0, 68.0, 46.0] Menu: 1=List comparison 2=Partial sum 3=Find max 0=Quit Enter your choice: 2 Compute sum of some days... Enter day number: 1 Enter day number: 4 Enter day number: 6 Enter day number: -99 Which teacher (1 or 2): 2 [1, 4, 6]

```
3=Find max
0=Quit
Enter your choice: 2
Compute sum of some days...
Enter day number: 1
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 2
[1, 3]
[40.0, 25.0, 15.5]
Sum of Days [1, 3] = 55.50
Continue (y/n): y

Compute sum of some days...
Enter day number: 2
Enter day number: -99
Which teacher (1 or 2): 1
[2]
[50.0, 20.0, 30.0]
Sum of Days [2] = 20.00
Continue (y/n): n

Menu:
1=List comparison
2=Partial sum
3=Find max
0=Quit
Enter your choice: 3
Which teacher (1 or 2): 1
Teacher1 spends maximum money = 50.00 on
Day1

Menu:
1=List comparison
2=Partial sum
3=Find max
0=Quit
Enter your choice: 0
```

```
[80.0, 124.0, 67.0, 20.75, 5.5, 33.0, 120.0, 71.0, 68.0, 46.0]
Sum of Days [1, 4, 6] = 133.75
Continue (y/n): y

Compute sum of some days...
Enter day number: 9
Enter day number: 10
Enter day number: -99
Which teacher (1 or 2): 1
[9, 10]
[100.0, 52.0, 84.0, 10.5, 20.25, 13.0, 157.0, 40.0, 68.0, 76.0]
Sum of Days [9, 10] = 144.00
Continue (y/n): y

Compute sum of some days...
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 1
[3]
[100.0, 52.0, 84.0, 10.5, 20.25, 13.0, 157.0, 40.0, 68.0, 76.0]
Sum of Days [3] = 84.00
Continue (y/n): n

Menu:
1=List comparison
2=Partial sum
3=Find max
0=Quit
Enter your choice: 1
Day 1: Teacher1 spends more than Teacher2
Day 2: Teacher1 spends less than Teacher2
Day 3: Teacher1 spends more than Teacher2
Day 4: Teacher1 spends less than Teacher2
Day 5: Teacher1 spends more than Teacher2
Day 6: Teacher1 spends less than Teacher2
Day 7: Teacher1 spends more than Teacher2
Day 8: Teacher1 spends less than Teacher2
Day 9: Teacher1 spends equal to Teacher2
Day 10: Teacher1 spends more than Teacher2

Menu:
1=List comparison
2=Partial sum
3=Find max
0=Quit
Enter your choice: 3
Which teacher (1 or 2): 1
Teacher1 spends maximum money = 157.00 on Day7

Menu:
1=List comparison
2=Partial sum
```

	3=Find max 0=Quit Enter your choice: 3 Which teacher (1 or 2): 2 Teacher2 spends maximum money = 124.00 on Day2 Menu: 1=List comparison 2=Partial sum 3=Find max 0=Quit Enter your choice: 0

Level 4 [0.5 Point]: (Save program in another file from Levels 1,2,3, filename = expense4.py)

Continue problem from Level 3. Add one more menu choice (choice 4) to get a partial list of expenses from specific days. See sample outputs for more understanding.

In your program, you must include the following 1 function:

* Your program can have more than this 1 function.

Function Name	Inputs	Outputs	Functionality
extract_list	A list of days	A list of expenses	Receive a list of specific days as input parameter. Form a new list of expenses from specific days and return this new list of expenses.

Sample outputs are shown below: *(Sample outputs only shows results for menu choice 4.)*

Level 4, Sample output 1:	Level 4, Sample output 2:
Enter your data set (A,B,C): A Teacher 1: [50.0, 20.0, 30.0] Teacher 2: [40.0, 25.0, 15.5] Menu: 1=List comparison 2=Partial sum 3=Find max 4=Extract list 0=Quit Enter your choice: 4 Get partial list of some days... Enter day number: 1 Enter day number: 3 Enter day number: -99 Which teacher (1 or 2): 2 [1, 3] [40.0, 25.0, 15.5] Partial list of Days [1, 3] = [40.0, 15.5]	Enter your data set (A,B,C): C Teacher 1: [100.0, 52.0, 84.0, 10.5, 20.25, 13.0, 157.0, 40.0, 68.0, 76.0] Teacher 2: [80.0, 124.0, 67.0, 20.75, 5.5, 33.0, 120.0, 71.0, 68.0, 46.0] Menu: 1=List comparison 2=Partial sum 3=Find max 4=Extract list 0=Quit Enter your choice: 4 Get partial list of some days... Enter day number: 1 Enter day number: 5 Enter day number: 9 Enter day number: 10 Enter day number: -99 Which teacher (1 or 2): 2 [1, 5, 9, 10]

```
Continue (y/n): y

Get partial list of some days...
Enter day number: 1
Enter day number: 2
Enter day number: -99
Which teacher (1 or 2): 1
[1, 2]
[50.0, 20.0, 30.0]
Partial list of Days [1, 2] = [50.0, 20.0]
Continue (y/n): y

Get partial list of some days...
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 1
[3]
[50.0, 20.0, 30.0]
Partial list of Days [3] = [30.0]
Continue (y/n): n

Menu:
1=List comparison
2=Partial sum
3=Find max
4=Extract list
0=Quit
Enter your choice: 0
```

```
[80.0, 124.0, 67.0, 20.75, 5.5, 33.0, 120.0, 71.0, 68.0, 46.0]
Teacher2: Partial list of Days [1, 5, 9, 10] = [80.0, 5.5, 68.0, 46.0]
Continue (y/n): y

Get partial list of some days...
Enter day number: 2
Enter day number: 3
Enter day number: 5
Enter day number: 8
Enter day number: 10
Enter day number: -99
Which teacher (1 or 2): 1

[2, 3, 5, 8, 10]
[100.0, 52.0, 84.0, 10.5, 20.25, 13.0, 157.0, 40.0, 68.0, 76.0]
Teacher1: Partial list of Days [2, 3, 5, 8, 10] = [52.0, 84.0, 20.25, 40.0, 76.0]
Continue (y/n): y

Get partial list of some days...
Enter day number: 3
Enter day number: -99
Which teacher (1 or 2): 2
[3]
[80.0, 124.0, 67.0, 20.75, 5.5, 33.0, 120.0, 71.0, 68.0, 46.0]
Teacher2: Partial list of Days [3] = [67.0]
Continue (y/n): n

Menu:
1=List comparison
2=Partial sum
3=Find max
4=Extract list
0=Quit
Enter your choice: 0
```