Array Assignments.

1. Write the code for the following problem. Assign 10 last names to an array. Write a function to display the names. Write another function to display the names in reverse order.

Input	Process	Output

Function	Def num (name):
	for n in name:
	print(n)
	
	def num2(name):
	name.reverse()
	for n in name:
	print(n)
		Function

Code	name = ["Ufir", "Tom", "Henry","Paul","Adam", "Sam","Tami","Andre","Quan","Jose"]	Names
Display	Names	Names & in reverse

2. Write the code for the following problem. Add another array to problem 1 above. This array should contain exam score for the respective students. That is, the first name goes with the first score etc. These are called parallel arrays. Also modify the display functions to include exam score array in addition to the last name array.

Input	Process	Output

Function	Def num (name):
	for n in name:
	print(n)
	def num2(name):
	name.reverse()
	for n in name:
	print(n)
	def score(exam,name):
	index = 0
	score = 0
	for e in name:
	score = exam[index]
	index += 1
	print("{} got {}".format(e, score))
		Function

Code	name = ["Ufir", "Tom", "Henry","Paul","Adam","Sam", "Tami","Andre","Quan"," Jose"]
	exam = ["90", "85", "89", "72", "75", "79", "83", "81", "70", "77"]
		Last name
		Exam Score

Display	Name and score	Name and score

3. Write the code for the following problem. The data to load is lastname and score. You can do this from a file. Add a function to problem to display the last name and highest, last name and lowest. Hint: for highest initialize a variable to 0 (high_var). If the array value is higher than the high_var then set high_var to the array value and set high_index to the position of the array. Proceed through the array until you get to the end. Do the same for finding the lowest using

low_var set to 999 (higher than the highest value).

Input	Process	Output

Function	def highLow(nameScore):
	global hindex
	global lindex
	inf = open('nameScore','r')
	content = inf.readlines()
	inf.close()
	for i in content:
	lst.append(i.rstrip('\n'))
	for i in lst:
	names.append(i[0:-2])
	scores.append(str(i[-2] + str(i[-1])))
	for i in scores:
	if int(i) < lowest:
	lowest == int(i)
	lindex = scores.index(i)
	elif int(i) > highest:
	highest == int(i)
	hindex = scores.index(i)
		Low and High

Input	Process	Output

code	highest = 0
	lowest = 999
	hindex = 0
	lindex = 0
	names = []
	scores = []
	lst = []
		Calls out high and low

Display	print('The highest score was {} by {}'.format(scores[int(hindex)],n ames[int(hindex)]])) print('The lowest score was {} by {}'.format(scores[int(lindex)],n ames[int(lindex)]]))	Print out the highest score and lowest score
File	Ufir 90 Tom 85 Henry 89 Paul 72 Adam 75 Sam 79 Tami 83 Andre 81 Quan 70 Jose 77	

4. Load list of 10 Player Names and Batting Averages from a file into arrays. (Create your own file

with two items: player last name and batting average, i.e. 0.267, 0.300 etc). Write a function to display the arrays. Then use a while loop to repeatedly ask the user for a last name. Write another function to search for the last name in the array and then display last name and batting average when found.

Input	Process	Output

Function	Def def
	highLow(NameAvg): inf
	= open('NameAvg','r')
	content = inf.readlines()
	inf.close()
	for i in content:
	lst.append(i.rstrip('\n'))
	for i in lst:
	names.append(i[0:-2])
	scores.append(str(i[-2] + str(I[-1])))	

File	Ufir 3 Tom .33 Henry .22 Paul 2.5 Adam 1.2 Sam .96 Tami 1.00 Andre .55 Quan 1.80 Jose 1.78	File for name and batting avg
Code	print(‘The Name: {} by {}'.format(name[int(lindex)],avg[i nt(lindex)]))	Print batting avg

Display		Batting and Average

5. Modify 4 above to display a message, “Name not found” when the name is not in the list. Example to be provided.

Load 10 employee last names and salaries into parallel arrays. Write a function to display the last names and salaries. Display the last names in reverse order. Write a function to find the employee with the highest salary. Write a loop to sum and display total of all salaries. Repeatedly ask the user for a name. Display the name and salary when found. Display message “Employee Not Found” when the last name is not in the list.

Input	Process	Output
File	Ufir 90000 Tom 80000 Henry 40000 Paul 35000 Adam 25000 Sam 75000 Tami 100000 Andre 55000 Quan 45000 Jose 60000	
Function	Lastn 0 1 2 3 Jonny, Joe, John, Josh Salary 0 1 3 4 1000, 9000, 8000, 5000	
Code		

Display		