

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.
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.
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
	<pre> Def displayname(lastn, scores): L = len(lastn) For y in range(0,l,1): Print(lastn[y], scores[y]) For x in range(l, -1, -1): Print(lastn[x], scores[x]) lastn=[] Scores=[] </pre>	
	<pre> Def hilow(lastn, scores): hiscore = -1.0 Lowscore = 99999999.99 For y in range(0,l,1): If float(scores[y] > float(hiscore): hiindex = y Lowscore = scores[y] If float(scores[y] < float(lowscore): Loindex = y Lowscore = scores[y] </pre>	<p>Display(lastn, scores)</p> <p>Hilow(lastn, scores)</p> <p>Last name + score</p> <p>Highest + Lowest Score + Lastname</p>

	<pre> Print"highest score", (lastn[hiindex], scores[hiindex]) Print(lowest score", lastn[loindex], scores[loindex]) Lastname = f.readline() While lastname != "": s = float(f.readline()) scores.append(s) Lastname = f.readline() f.close() </pre>	
--	---	--

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.
5. Modify 4 above to display a message, "Name not found" when the name is not in the list.

Input	Process	Output
	<pre> Def displaynames(lastn, batting) L=len(lastn) For x in range(1,l,1): Print(x," ", lastn[x], " ", batting[x]) F= open("lname", "r") While lastname != "": B = float(f.readline()) Batting.append(B) Lastname = f.readline() f.close() </pre>	

Response sname	While reponse == "yes" Sname = input("Enter lastname") Def seqsearch (lastn, sname): l = seqsearch(lastn, sname) If l == 1" Print(sname, "not found") Else: Print(lastn[l], "average of", batting[l])	Lastnames Battingaverages
		Display (lastname, batting)

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.