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 |
| Last name | def display (last name):  for i in range (0, 10, 1)  print (last name [i])  def reverse (last name):  for x in range (9, -1, -1)  print (last name [x])  display(last name)  print("\n")  reverse(last name) | Last name  Reverse name |
|  |  |  |

1. 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 |
| Last name | def exam scores (last name, def display exam scores (last name, scores):  for x in range (0,10,1): print(last name[x], "scores", scores[x])  display exam scores (last name, exam scores) | Last name  Scores |
| Exam scores |  |  |
|  |  |  |

1. 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 |
| Last name | f = open("Code/PS13P3(11).txt", "r")  lastn = []  score = []  lastname = f.readline() | Last name with high score  Last name with low score |
| scores | while lastname != "":  lastn.append(lastname.rstrip("\n"))  scores = float(f.readline())  score.append(scores)  lastname = f.readline() |  |
|  | f.close()  l = len(lastn) |  |
|  | def low(lastn, score):  low\_var = 999  lowindex = 0  for y in range(l):  if score[y] < low\_var:  lowindex = y  low\_var = score[y]  print(f"{lastn[lowindex]} has the lowest score of {score[lowindex]}.") |  |
|  | def high(lastn, score):  high\_var = -1  highindex = 0  for x in range(l):  if score[x] > high\_var:  highindex = x  high\_var = score[x]  print(f"{lastn[highindex]} has the highest score of {score[highindex]}.") |  |
|  | print(lastn, score)  high(lastn, score)  low(lastn, score) |  |
|  |  |  |

1. 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 |
| 10 player names | f = opne (“file”, “r”)  player names = []  batting averages = []  Player = f. readline () | Searched player names  Batting average |
| Batting averages | While player ! = “ ”:  Player names, append (str (player). rstrip (“\n”))  ba= float (f. readline())  Batting averages. Append (ba)  Playernames = f. readline()  I = len (playernames) |  |
|  | Def display (player names, batting averages):  For z in player names, batting averages:  Print (z) |  |
|  | Def search (player names, player, batting averages, playersearch)  For y in range (0, 9, 1):  If playernames [y] in. playersearch:  Sindex = y  Print (player names[sindex]," the batting average is",batting averages[sindex])  Display (player, batting averages) |  |
|  | User’s choice= Do you want to search for a player? (yes/no)  While user’s choice == “yes”:  playersearch=str(input("What player do you want to search?"))  search(playernames,player,batting avarage,playersearch) |  |
|  | User’s choice = str(input("Do you want to search another player? ('Yes' or 'No')")) |  |
|  |  |  |

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

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 10 player names | f = opne (“file”, “r”)  player names = []  batting averages = []  Player = f. readline () | Searched player names  Batting average |
| Batting averages | While player ! = “ ”:  Player names, append (str (player). rstrip (“\n”))  ba= float (f. readline())  Batting averages. Append (ba)  Playernames = f. readline()  I = len (playernames) |  |
|  | Def display (player names, batting averages):  For z in player names, batting averages:  Print (z) |  |
|  | Def search (player names, player, batting averages, playersearch)  For y in range (0, l, 1):  If playernames [y] in. playersearch:  Sindex = y  Print (player names[sindex]," the batting average is",batting averages[sindex])  Return  print("Player is not found.")  Display (player, batting averages) |  |
|  | User’s choice= Do you want to search for a player? (yes/no)  While user’s choice == “yes”:  playersearch=str(input("What player do you want to search?"))  search(playernames,player,batting avarage,playersearch) |  |
|  | User’s choice = str(input("Do you want to search another player? ('Yes' or 'No')")) |  |
|  |  |  |

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.