

I included the original text file that I used before I ran this for testing purposes, you can run what I ran and get the end version of the text file. I overloaded the comments. Dau gives good advice. Takes me a while haha.

# Robert Cyril Plata III  
# Sally Kyvernitis  
# 10/29/2015  
# Only enter capital team names for consistency, will not allow lower. Enter capital teams for insert.  
# Case sensitive program "E" not "e"  
# Features - Print straight from the text file - Print the list version - Search for a key  
# Remove a key and parallel counterparts - Insert a new key with parallel wins/loss - Leave and overwrite the text file  
# This is my UI  
**def menu(**prompt**):** # prompt is below in the main program, it just asks choose an option.  
 running **= True** # loop until false, user can type in wrong option and it will go until good option is fulfilled.  
 print**("Enter 1 : Print (text file)")** # User sees text  
 print**("Enter 2 : Search")** # User sees text  
 print**("Enter 3 : Insert")** # User sees text  
 print**("Enter 4 : Delete")** # User sees text  
 print**("Enter 5 : PrintList (List)")** # User sees text  
 print**("Enter 6 : Leave and Save")** # User sees text  
 **while** running**:** # Will run until turned False  
 userInput **=** input**(**prompt**)  
 if** userInput **== "1":** # if option then execute code below.  
 userInput **= "Print"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **elif** userInput **== "2":** # if option then execute code below.  
 userInput **= "Search"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **elif** userInput **== "3":** # if option then execute code below.  
 userInput **= "Insert"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **elif** userInput **== "4":** # if option then execute code below.  
 userInput **= "Delete"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **elif** userInput **=="5":** # if option then execute code below.  
 userInput **= "PrintList"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **elif** userInput **== "6":** # if option then execute code below.  
 userInput **= "Leave"** # Changed value for simplicity. It matches the main programs options below.  
 running **= False** # Turned False  
 **else:** # else for wrong input  
 print**()** print**("Sorry,"**, userInput, **"is not an option.")** # notifying user, choice not valid  
 print**()  
 return** userInput # returns the final choice (userInput) main program will now match accordingly  
  
**def defineLists():  
 for** lineOfText **in** Teams**:** # for potato in Teams, (Teams is the openFile), do the following below, until complete! potato = lineOfText ;D  
 name **=** lineOfText**[**START**:**NAME\_LEN**]** # length that i want an array  
 name **=** name.strip**()** # removing unnecessary spaces  
 teamName.append**(**name**)** # Add line to list teamName until completion  
 wins **=** lineOfText**[**NAME\_LEN**:**WIN\_LEN**]** # length i want the second array  
 wins **=** wins.strip**()** # removing unnecessary spaces  
 teamWins.append**(**wins**)** # Add line to list teamWins until completion  
 loss **=** lineOfText**[**WIN\_LEN**:**LOSS\_LEN**]** # length wanted for third array  
 loss **=** loss.strip**()** # removing unnecessary spaces  
 teamLoss.append**(**loss**)** # Add line to list teamLoss until completion  
 **return (**name, wins, loss**)** # return in case needed.  
  
# Prints List so that way the user can see what the list looks like  
**def printList():** print**()  
 for** i **in** range**(**0,len**(**teamName**)):** # prints each line neatly until completion  
 **if** i **>=** 10**:** # When i reaches 10 it will now take away a little bit of space so that way everything lines up. Not completely necessary, but for the eyes.  
 print**(**i, format**(**teamName**[**i**]**, NAME\_FORMAT**)**, **" "**, format**(**teamWins**[**i**]**, WIN\_FORMAT**)**.strip**()**,**" "**,format**(**teamLoss**[**i**]**, LOSS\_FORMAT**)**.strip**())  
 if** i **<** 10**:** # Before i reaches 10, it is the normal format.  
 print**(**i, **""**, format**(**teamName**[**i**]**, NAME\_FORMAT**)**, **" "**, format**(**teamWins**[**i**]**, WIN\_FORMAT**)**.strip**()**,**" "**,format**(**teamLoss**[**i**]**, LOSS\_FORMAT**)**.strip**())** print**()** print**("Team /"**, **"Wins /"**, **"Losses /"**, **"index")** # Organization, what each column means  
 print**()**# Prints straight from the text file if the user wants to see what they started with  
**def printFile(**teams**):  
 for** lineOfText **in** teams**:** # Prints each line neatly until completion  
 lineOfText **=** lineOfText.strip**()** # strip for spaces  
 teamNames **=** lineOfText**[**START**:**NAME\_LEN**]** # print the range of this list  
 record **=** lineOfText**[**NAME\_LEN**:**LOSS\_LEN**]**# print the range of this list  
 record **=** record.strip**()**# print the range of this list  
 print**(**teamNames**+**format**(""+**record, **'^10s'))** # formating the files output on program  
 print**()** print**("Printing Successful!")  
  
def searchList(**searchName**):** # Searches the list and returns the parallel equal  
 **for** i **in** range **(**START, len**(**teamName**)):  
 if** teamName**[**i**]**.startswith**(**searchName**):** # if something in teamName starts with the users inputted letter, it will return the location  
 #print (i) useful for debugging  
 **return** i # Returns number(location) of the found searchName  
 **return -**1 # returns -1 so the insert function sees that the team doesn't exist yet. When the if loop doesn't find anything it will return -1  
   
**def insert(**team**):** # team is the input arguement the user searched in the main program, whether it exists or not, it is relevant to this function  
 **if** team **not in** teamName**:** # if the team arguement is not inside of the list teamName, then it will ask for wins and losses for the new Team that the user entered in the main program  
 wins **=** input**("Enter Wins:")** # user inputs wins  
 loss **=** input**("Enter losses:")** # user inputs losses  
 **for** i **in** range **(**0,len**(**teamName**)):** # for loop   
 **if** teamName**[**i**]>**team**:** # for alphabetical order insertion, if teamName[i] is greater than team alphabetically  
 teamName.insert**(**i, team**)** # inserts team into location i of list teamName  
 teamWins.insert**(**i, wins**)** # inserts wins into location i of list teamName  
 teamLoss.insert**(**i, loss**)** # inserts loss into location i of list teamName  
 **return** # Returns nothing so that way it stops the function. This is actually vital to the program inserting. Without it, this part of the function will not complete.  
 **if** teamName**[**i**]<**team**:** # Another option in case someone enters a key less than the last key alphabetically  
 teamName.insert**(**len**(**teamName**)**, team**)** # inserts team into location i of list teamName  
 teamWins.insert**(**len**(**teamName**)**, wins**)** # inserts wins into location i of list teamName  
 teamLoss.insert**(**len**(**teamName**)**, loss**)** # inserts loss into location i of list teamName  
  
 **elif** team **in** teamName**:** # if team is in teamName because it skipped the first 'if' then it will tell the user that team exists already and that they can edit the record.  
 print**(**team, **"exists, edit the record.")** wins **=** input**("Enter Wins:")** # User enters the wins  
 loss **=** input**("Enter losses:")** # User enters the losses  
 **for** i **in** range**(**START,len**(**teamName**)):** # This for loop will search from 0 to the length of the list  
 **if** teamName**[**i**]==**team**:** # if teamName matches the users input of team in the main program, it will stop the for loop and change the values below.  
 teamWins**[**i**] =** wins # teamWins[i] is the exact win number of that particular parallel list point, it will then change it to equal the new value asked above.  
 teamLoss**[**i**] =** loss # This does the exact same thing ^^ it will change the value to the one above.  
   
# No longer used because I don't need it.  
**def clearScreen():** #the function with no arguement  
 **for** i **in** range**(**START, 40**):** # for loop from 0 to 40  
 print**()** # just print blank lines  
  
# Deletes a string in the list  
**def deleteItem(**deleteElement**):** # deleteElement (input arguement) is the element the user wants to remove.  
 **for** i **in** range**(**deleteElement, len**(**teamName**)-**1**):** # for loop from the users input to the length of teamName -1 because we are removing something. We do not want to get an overage error.  
 teamName**[**i**] =** teamName**[**i**+**1**]** # teamName[i] is now +1 because we are going to pop.  
 teamName.pop**()** # last part of list removed.  
  
# Constants  
START **=** 0 # Start is a 0 constant.  
NAME\_LEN **=** 9 # Name len is the amount of characters we will go up to, to define the parallel list teamNames  
WIN\_LEN **=** 18 # Win len will be the area in which the wins are in  
LOSS\_LEN **=** 23 # Loss will cut off right after the very end where losses are in the text file  
NAME\_FORMAT **=** str**(**NAME\_LEN**)** # the string of value concatenated with "s" for string formation for will make formatting easier without using magic numbers.   
WIN\_FORMAT **=** str**(**WIN\_LEN**)+"s"** # same ^^  
LOSS\_FORMAT **=** str**(**LOSS\_LEN**)+"s"** # same ^^  
WRITE\_FORMAT1 **=** str**('>7s')**WRITE\_FORMAT2 **=** str**('>6s')**# Defining Lists  
Teams **=** open**("TeamsAndRecords.txt"**,**"r")** # will open the file for reading, it is now represented as Teams  
teamName **= []** # List defining  
teamWins **= []** # List defining  
teamLoss **= []** # List defining  
  
defineLists**()** #defining all of the lists that are needed. There will now be 3 parallel defined from the file represented as Teams.  
   
ON **= True** # boolean is true  
**while** ON**:** #boolean, ON is True  
 Menu **=** menu**("Choose an option.")** # Menu is func menu  
  
 # Print option  
 **if** Menu **== "Print":** # Menu = func menu and it returned "Print"  
 Teams **=** open**("TeamsAndRecords.txt"**,**"r")** print**(**format**(" Football Teams")**,**)** # Title  
 print**()** print**("Teams Wins Losses")** # Labels  
 print**("------------------------")** # Border  
 printFile**(**Teams**)** # Initiate printFile function  
  
 # Search option  
 **elif** Menu **== "Search":** # Menu = func menu and it returned "Search"  
 askSearch **=** input**("What are you looking for?")** index **=** searchList**(**askSearch**)** # users input is searched  
 **if** index **>=** 0**:** # If the user enters a number less than zero throw exception  
 print**()** print**("Found:"**, teamName**[**index**]**, **"with a record of"**, teamWins**[**index**]**, **"and"**, teamLoss**[**index**])** print**()  
 else:** print**()** print**("not found")** print**()** # Insert option  
 **elif** Menu **== "Insert":** # Menu = func menu and it returned "Insert"  
 entryTeam **=** input**("What Team?")** # variable for user input of the team they enter  
 searchInsert **=** searchList**(**entryTeam**)** # search the variable, if found it will return  
 insert**(**entryTeam**)** # calls function with entryTeam arguement, it will insert if -1 was returned from searchInsert. if 0 or above was returned, it will allow user to change value of win and loss  
  
  
 # Delete option  
 **elif** Menu **== "Delete":** # Menu = func menu and it returned "Delete"  
 element **=** input**("What team do you want to remove?")** searchDelete **=** searchList**(**element**)  
 if** element **in** teamName**:** # if element is in the list then..  
 deleteItem**(**searchDelete**)** #delete that searched element  
 print**(**element, **"deleted")** # Prints the user's new list  
 print**()  
 else:** # if element was not in the list..  
 print**()** print**(**element, **"Not in list")** # element wasn't in the list  
 print**()** # Simple PrintList option  
 **elif** Menu **== "PrintList":** # Menu = func menu and it returned "PrintList"  
 printList**()** # printList func called  
  
 # Leave option  
 **elif** Menu **== "Leave":** # Menu = func menu and it returned "Leave"  
 print**()** print**("Final list is:")** printList**()** # prints the final list one last time  
 print**()** print**("G")** print**(" O")** print**(" O")** print**(" D")** print**(" B")** print**(" Y")** print**(" E")** print**(" !!")** # message for leaving  
  
 # Beginning process of writing to original text file  
 inFile **=** open**("TeamsAndRecords.txt"**, **"w")** #open TeamsAndRecords for overwriting  
 **for** i **in** range**(**START,len**(**teamName**)):** # for loop to write to inFile from 0 to the length of list teamName  
 inFile.write**(**format**(**teamName**[**i**]**,NAME\_FORMAT**))** # write the names in teamName list to the document making sure it is formatted correctly  
 inFile.write**(**format**(**teamWins**[**i**]**,WRITE\_FORMAT1**))** # write wins in teamWins using format function to line up where it first started  
 inFile.write**(**format**(**teamLoss**[**i**]**,WRITE\_FORMAT2**))** # write losses in teamLoss using format function as well  
 inFile.write**("\n")** # '\n' will created a new line in my file  
 inFile.close**()** # close, without closing the file we are writing to, it will not save.  
 print**()** print**("Overwrite Successful")** ON **= False** # boolean ON is false