Define two lists at the top of your file such as the ones below:

names = ["Aliyah", "Bob", "Cathy", "Dan", "Ed", "Frank", "Darnell",   
 "Gary", "Shanice", "Irene", "Jack", "Kelly", "Demetrius"]  
ages = [20, 21, 18, 18, 19, 20, 20, 19, 19, 19, 22, 19, 30]

These lists should match up, so Aliyah's age is 20, Bob’s age is 21, and so on. In this example, the names should be the keys and the age should be the value.

* Use the zip function to merge these lists into a dictionary.
  + Include a comment - What data type does zip() return?
  + Include a comment - How do you coerce that to the right data type?
* Give a the user 5 tries to check to see if a name is in the dictionary
  + give them up to five tries by using a sentinel variable
  + use a while loop to continue to ask if they haven't found a name in the dictionary or until they run out of tries
  + ask the user to input a name
  + check if the name is in the dictionary (hint: use [membership](https://canvas.uw.edu/courses/1347250/pages/dictionaries#dictionary-membership) or [get](https://canvas.uw.edu/courses/1347250/pages/dictionaries#dictionary-get))
    - if the user is in the dictionary, return the user's age
    - if the user is not in the dictionary, return an error message

Your program should print out the response, as follows:

"Please input an user to find out their age: "  
"Aliyah"   
"Aliyah is 20!"   
  
"Please input an user to find out their age: "  
"Tabitha"  
"There is nobody here named Tabitha, please try again: "

Review:

[Lists To Dictionaries](https://canvas.uw.edu/courses/1347250/pages/dictionaries#lists-zip-dictionaries)

[While Loop with Limited Tries](https://canvas.uw.edu/courses/1347250/pages/handling-user-input#limited-tries)

[Dictionary membership](https://canvas.uw.edu/courses/1347250/pages/dictionaries#dictionary-membership)

[Dictionary get](https://canvas.uw.edu/courses/1347250/pages/dictionaries#dictionary-get)

**Rubric**

Assignment 04 (2)

| Assignment 04 (2) | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning Outcome Script properly combines two lists |  | 15.0 pts |
| This criterion is linked to a Learning Outcome while loop allows user to guess 5 names |  | 15.0 pts |
| This criterion is linked to a Learning Outcome script gets user input with proper error handling |  | 15.0 pts |
| This criterion is linked to a Learning Outcome Script prints out correct names and age corresponding to input value. |  | 15.0 pts |
| This criterion is linked to a Learning Outcome if name is not in dictionary, script prints an error message |  | 15.0 pts |
| This criterion is linked to a Learning Outcome script runs without errors and code is documented |  | 15.0 pts |
| Total Points: 90.0 | | | |