Instructions – A3 – Friend Tracker

Overview:
You'll be creating a simple python program that allows users to manage a directory of their friends' hobbies. The program will prompt the user to add new friends with their corresponding hobbies and also allow them to look up a friend's hobby.

Logical Flow:

Print out a menu that gives the user 3 options:

- "Menu:"
- "1. Add a Friend"
- "2. Find a Friend's Hobby"
- "3. Quit"

Then ask for input from the user to enter 1, 2, or 3.

If the user enters "1":

Prompt the user to enter:

- A friend's name.
- That friend's hobby.

Then store that data in a dictionary, with the friend's name as the key and the hobby as the value, and print out "{friend name} added to your dictionary!".

But, before you add the friend/hobby to the dictionary, check to see if that friend is already in the dictionary. If they are, don't try to add them, but instead print out "{friend name} is already in your dictionary." There are a few ways to do this. An if statement, or doing something with the .get() function are a couple different ways to go about it.

If the user enters "2":

Prompt the user "Enter friend's name to find their hobby: "

After the user enters a name, print out :"{name}'s hobby is {hobby}." If the friend's name is in the dictionary. If it isn't in the dictionary, print out "{name} is not in the dictionary."

If the user enters "3":

Print out ""Exiting the program. Goodbye!"

If the user enters anything other than 1, 2, or 3:

Print out "Invalid choice. Please choose a valid option."

The menu with the 3 choices should continually reappear every time choice 1, 2 (or an invalid input) is entered, meaning the user can input as many friends/hobbies as they want, and search for friends' hobbies. The program should only end if the user enters 3.

Upload just the python file to Learning Suite. This means you should upload the .py file that you made. You will lose points if you copy/paste your code, or upload something like a word file instead.

Example Output:

```
Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): 1
Enter friend's name: Amir
Enter Amir's hobby: Baking

Amir added to your dictionary!

Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): 1
Enter friend's name: Sarah
Enter Sarah's hobby: snowboarding

Sarah added to your dictionary!

Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): 2
Enter friend's name to find their hobby: Sarah

Sarah's hobby is snowboarding.

Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): 2
Enter friend's name to find their hobby: Jimmy

Jimmy is not in the dictionary.

Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): example of invalid input

Invalid choice. Please choose a valid option.
```

Menu:
1. Add a Friend
2. Find a Friend's Hobby
3. Quit
Choose an option (1, 2, or 3): 3

Exiting the program. Goodbye!

Rubric:

Requirement	Points	Notes
prints out the menu with 3 options	5	
asks for user to enter 1, 2, or 3	5	
Allows entering in a new friend/hobby	20	· -7 if the name or hobby incorrectly stored in a dictionary · -5 if it doesn't print out message for case when the name is already in the dictionary ·-3 if it doesn't include the name in the message(s) like "{name} added to your dictionary!"
Allows searching for a hobby by friend name	20	· -7 if the correct hobby doesn't show up for an entered name. · -5 if it doesn't print out message for case when the name is not in the dictionary ·-3 if it doesn't include the name in the message(s) like "{name} is not in the dictionary!"
Allows option for exiting the program	20	· -7 if the program doesn't end when choosing this option · -5 if it doesn't print out message "Exiting the program. Goodbye"
Prints out a message for incorrect input	5	· -5 if it does not print out message when anything other than 1, 2, or 3 is entered.
continually shows menu and allows for more input after 1, 2, or invalid input is entered.	20	· -10 if they attempt it but it doesn't work properly.
includes comments: name and description at the top, and comments throughout	3	· -1 if it didn't include their name · -1 if it didn't include a project description · -2 if it didn't include general comments. Err on the side of leniency
file is turned in as a .py file	2	
Total	100	