Think of at least five places in the world you'd like to visit.

- Store the locations in a list. Make sure the list is not in alphabetical order.
- Print your list in its original order. Don't worry about printing the list neatly, just print it as a raw Python list.
- Use sorted() to print your list in alphabetical order without modifying the actual list.
- Show that your list is still in its original order by printing it.
- Use sorted() to print your list in reverse alphabetical order without changing the order of the original list.
- Show that your list is still in its original order by printing it again.
- Use reverse() to change the order of your list. Print the list to show that its order has changed.
- Use reverse() to change the order of your list again. Print the list to show it's back to its original order.
- Use sort() to change your list so it's stored in alphabetical order. Print the list to show that its order has been changed.
- Use sort() to change your list so it's stored in reverse alphabetical order. Print the list to show that its order has changed.

4-1. Pizzas: Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a for loop to print the name of each pizza.

- Modify your for loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like I like pepperoni pizza.
- Add a line at the end of your program, outside the for loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as I really love pizza!

4-2. Animals: Think of at least three different animals that have a common characteristic. Store the names of these animals in a list, and then use a for loop to print out the name of each animal.

- Modify your program to print a statement about each animal, such as A dog would make a great pet.
- Add a line at the end of your program stating what these animals have in common. You could print a sentence such as Any of these animals would make a great pet!

Use a for loop to print the numbers from 1 to 20, inclusive.

4-4. One Million: Make a list of the numbers from one to one million, and then use a for loop to print the numbers. (If the output is taking too long, stop it by pressing ctrl -C or by closing the output window.)

4-5. Summing a Million: Make a list of the numbers from one to one million, and then use min() and max() to make sure your list actually starts at one and ends at one million. Also, use the sum() function to see how quickly Python can add a million numbers.
4-6. Odd Numbers: Use the third argument of the range() function to make a list of the odd numbers from 1 to 20. Use a for loop to print each number.
4-7. Threes: Make a list of the multiples of 3 from 3 to 30. Use a for loop to print the numbers in your list.
4-8. Cubes: A number raised to the third power is called a cube. For example, the cube of 2 is written as 2**3 in Python. Make a list of the first 10 cubes (that is, the cube of each integer from 1 through 10), and use a for loop to print out the value of each cube.
4-9. Cube Comprehension: Use a list comprehension to generate a list of the first 10 cubes
----

Buffet-style restaurant offers only five basic foods. Think of five simple foods, and store them in a tuple.
• Use a for loop to print each food the restaurant offers.
• Try to modify one of the items, and make sure that Python rejects the change.
• The restaurant changes its menu, replacing two of the items with different foods. Add a block of code that rewrites the tuple, and then use a for loop to print each of the items on the revised menu.

5-8. Hello Admin: Make a list of five or more usernames, including the name 'admin' . Imagine you are writing code that will print a greeting to each user after they log in to a website. Loop through the list, and print a greeting to each user:
• If the username is 'admin' , print a special greeting, such as Hello admin, would you like to see a status report?
• Otherwise, print a generic greeting, such as Hello Eric, thank you for logging in again.

5-9. No Users: Add an if test to hello_admin.py to make sure the list of users is not empty.
• If the list is empty, print the message We need to find some users!
• Remove all of the usernames from your list, and make sure the correct message is printed.

5-10. Checking Usernames: Do the following to create a program that simulates how websites ensure that everyone has a unique username.
• Make a list of five or more usernames called current_users .
• Make another list of five usernames called new_users . Make sure one or two of the new usernames are also in the current_users list.
• Loop through the new_users list to see if each new username has already been used. If it has, print a message that the person will need to enter a

new username. If a username has not been used, print a message saying that the username is available.

- Make sure your comparison is case insensitive. If 'John' has been used, 'JOHN' should not be accepted.

5-11. Ordinal Numbers: Ordinal numbers indicate their position in a list, such as 1st or 2nd. Most ordinal numbers end in th, except 1, 2, and 3.

- Store the numbers 1 through 9 in a list.
- Loop through the list.
- Use an if - elif - else chain inside the loop to print the proper ordinal ending for each number. Your output should read "1st 2nd 3rd 4th 5th 6th 7th 8th 9th" , and each result should be on a separate line.