

6.6 Exercises

Write a complete Python program to solve each of the following problems.

- 6.1. Write a program that reads a list of numbers from the user; the list terminates when the user enters 0. After all of the input the program should print the list.
- 6.2. Change the program from problem 6.1 so the list is printed in reverse order.
- 6.3. Change the program from problem 6.1 to use a tuple instead of a list.
- 6.4. Change the program from problem 6.1 to print the average value of the list as well as each of the values in the list.
- 6.5. Write an inventory program for a store, similar to program 6.3.1. This should let the user enter objects and the number of each in the inventory. The input terminates when the user gives the empty string as an object name. After this the program prints the entire inventory, alphabetized by the names of the objects. For example, the input might be

Object: widget
Number of widgets: 24

Object: thingamabob
Number of thingamabobs: 3

Object: gizmo
Number of gizmos: 15

Object:

The output will then be:

gizmos: 15
thingamabobs: 3
widgets: 24

Make the user interaction as natural as you can. Note the use of singular and plural objects in this example.

- 6.6. Change the program from problem ?? so that the output is in order from smallest number to largest number. For the example given it will be

thingamabobs: 3
gizmos: 15
widgets: 24

- 6.7. Write a program that inputs a list of values (it doesn't matter if they are numbers or strings) and then prints all of the values that appear in the list more than once.
- 6.8. Write a program that inputs a list of numbers and outputs basic statistical information about the list: minimum and maximum values, average value, standard deviation (the square root of variance; you can find this in any book on statistics), and median (middle value).
- 6.9. Write a program that stores in a dictionary a list of people's birthdays. To make the input simple, read in the three fields of the birthday (month, day, year) on separate lines. Your input might look like this:

```
Enter name: bob  
Enter month: 2  
Enter day: 16  
Enter year: 1952  
Enter name:
```

It this could keep up until you get a blank name. With each name you should enter the name and date in a dictionary, with the name as the key, and a tuple for the date as a value. After all of the input, have your program print all of the birthdays in the dictionary.

- 6.10. Add a `LookupName` feature to the program from 6.9. This should ask the user for a name, and then print the birthday for that person. Make sure your program doesn't crash if you give it the name of someone who isn't in the dictionary.
- 6.11. Add a `LookupBirthday` feature to the program from 6.9. This should ask for a birthday, and then print all of the people who had that birthday.
- 6.12. Try rewriting the program from problem 6.9 so that the keys of the dictionary are the dates (use a tuple `(m, d, y)` for this) and the values are lists of all people with that date as their birthday.