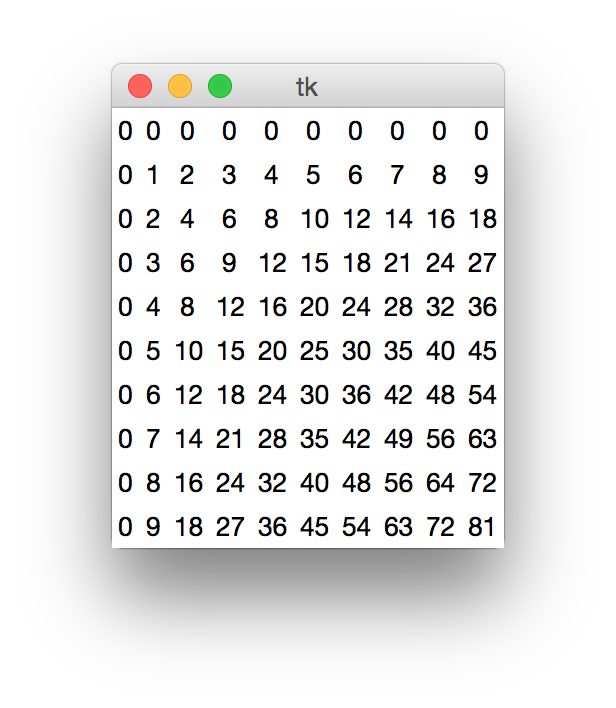
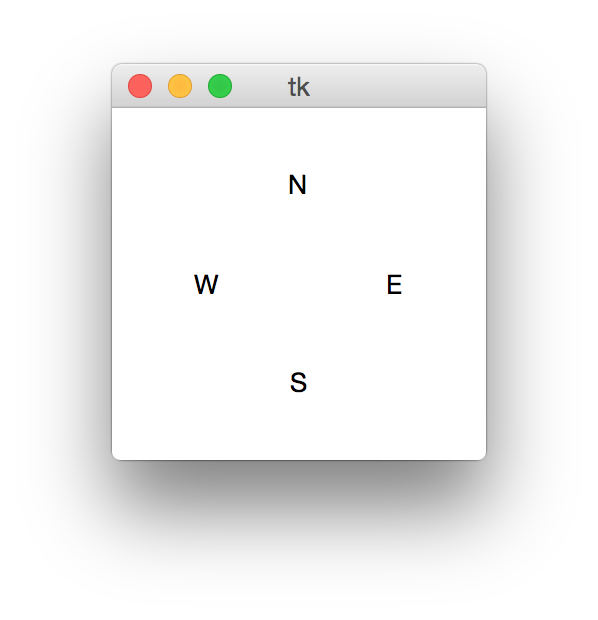
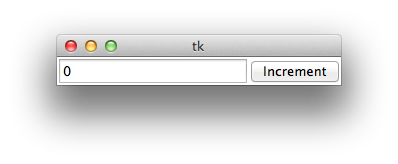
**Chapter 9 test problems**

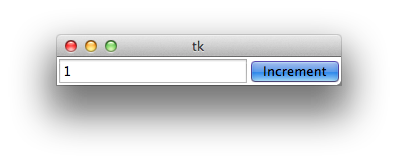
1. Implement the GUI widget class that displays a multiplication table as shown below. Note: the table is a 10x10 grid.



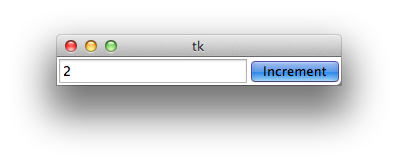
2. Implement GUI widget class that contains 4 Label widgets corresponding to the 4 directions on a compass, as shown below.



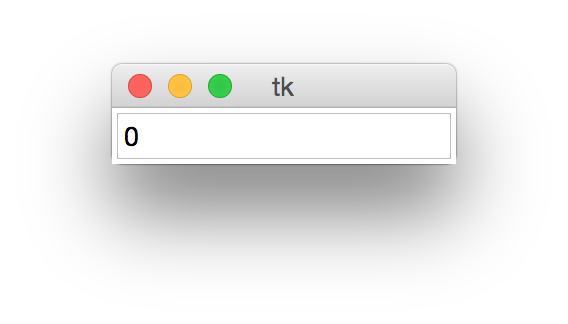
3. Implement a GUI widget class that consists of an Entry widget and a Button widget labeled "Increment". The Entry widget should contain the number 0 at startup:  
  
When the user clicks on the Button, the number in the Entry should be incremented:



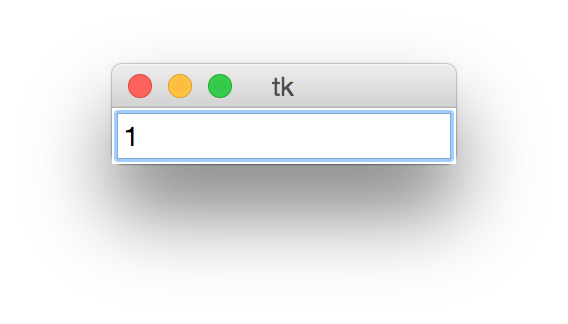
Additional clicks on the Button should continue to increment the value in the Entry:



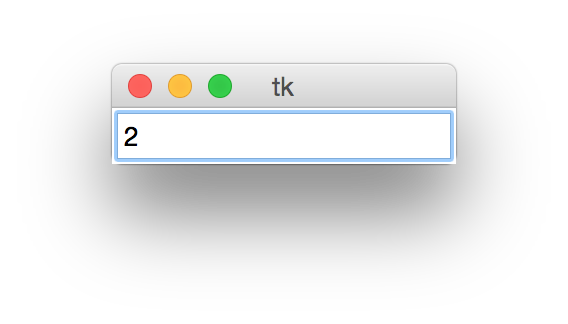
4. Implement a GUI widget class that consists of an Entry widget that contains the number 0 at startup:

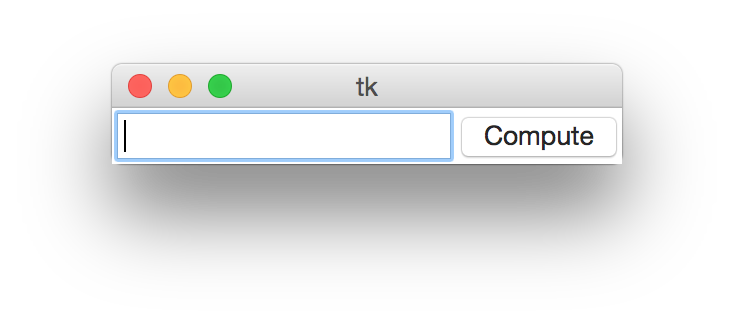


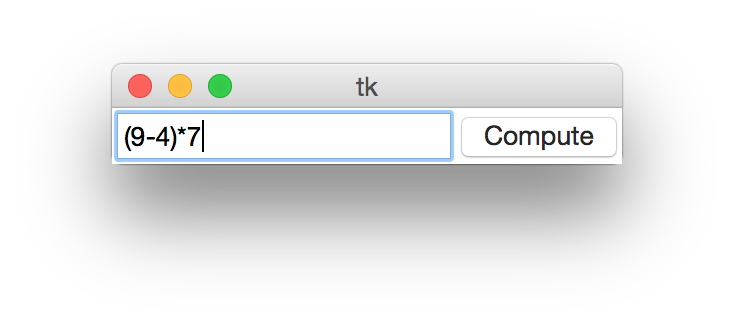
When the user presses on the Enter/Return key on the keyboard, the number in the Entry should be incremented:

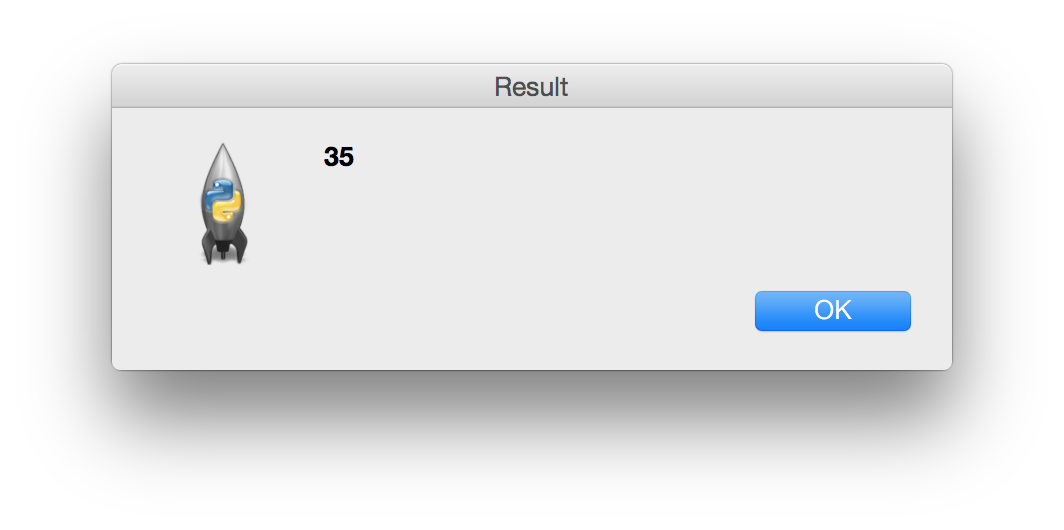


Additional presses on the Enter/Return key should continue to increment the value in the Entry:

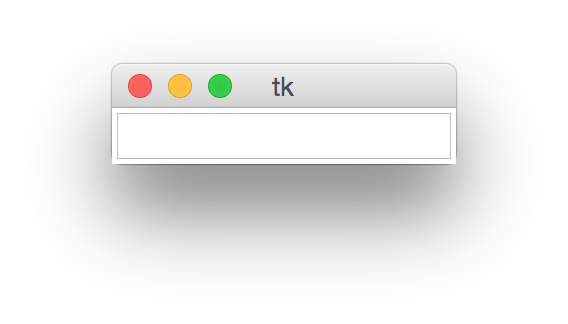


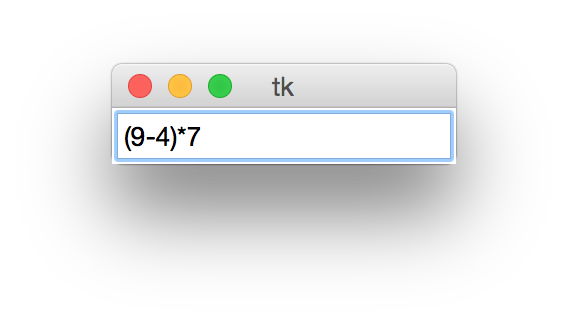
5. Implement a GUI widget class Calculator that consists of an Entry widget and a Button widget labeled "Compute":  


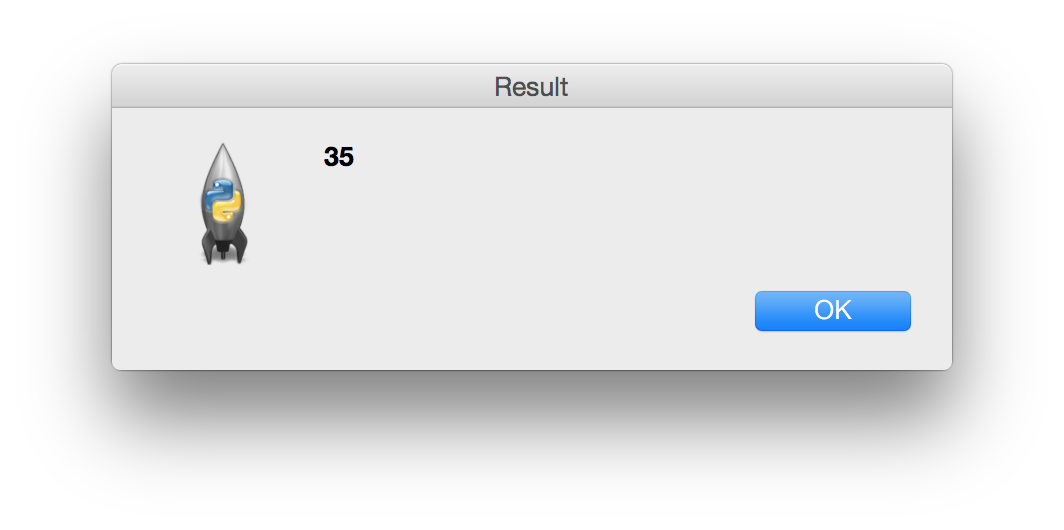
To use the app, the user should enter an arithmetic expression in the Entry:  


When the user clicks on the Button, a popup window should display the result:  


After the user clicks OK in the popup window, the Entry should be empty and ready for the next expression (as shown in the first image).

6. Implement a GUI widget class Calculator that consists of an Entry widget and a Button widget labeled "Compute":  


To use the app, the user should enter an arithmetic expression in the Entry:  


When the user clicks on the Button, a popup window should display the result:  


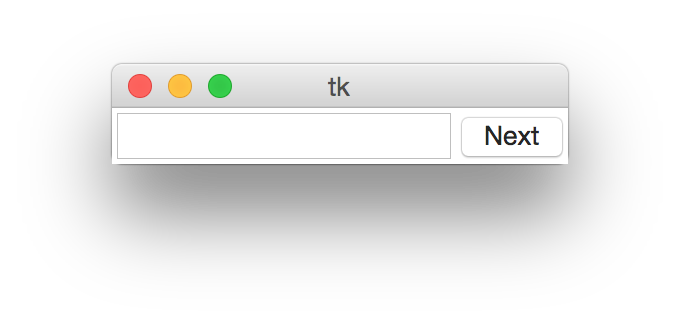
After the user clicks OK in the popup window, the Entry should be empty and ready for the next expression (as shown in the first image).

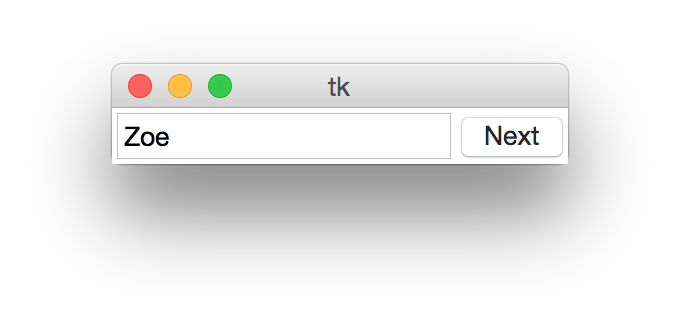
7. Implement a GUI widget class RoundRobin that consist of an Entry widget and a Button widget labeled "Next". The class constructor should take a non-empty list as an input argument, for example:

>>> lst = ['Zoe', 'Yannick', 'Xena', 'Wendy', 'Vince']

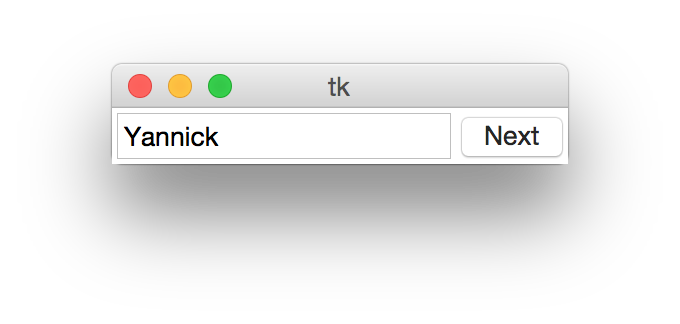
>>> RoundRobin(lst).mainloop()

The Entry widget should display nothing at startup:

  
When the “Next” button is clicked, the first item in the list should be displayed:

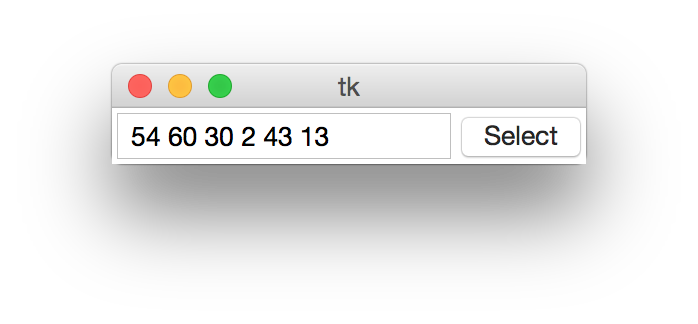


Clicking the “Next” button again should result in the next list item being displayed:

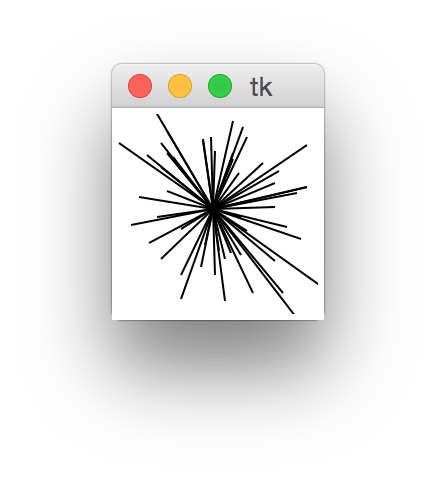


In general, every button click should result in the next list item being displayed, in round robin fashion.

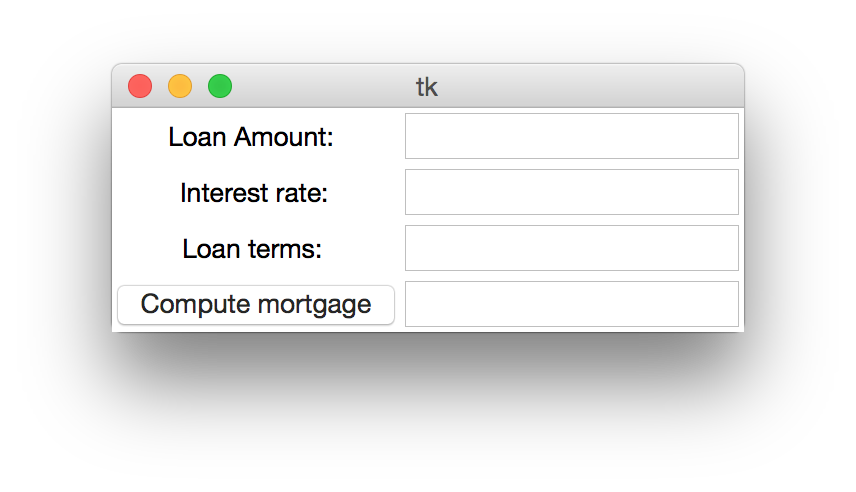
8. Implement a GUI widget class Lottery that consists of an Entry widget and a Button widget labeled "Select" and that implements a lottery numbers selection tool. When started, the Entry widget should be empty. Each button click should add a number selected without replacement from the positive integers in the range from 1 to 60. After 6 button clicks, the GUI would look something like:

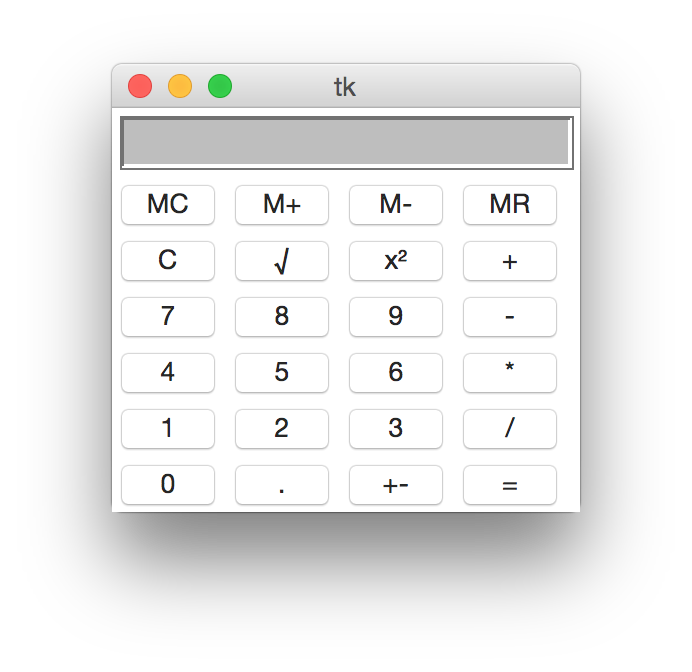


9. Implement a GUI widget class Star that contains a 100x100 Canvas widget. Your widget should behave as follows: When the mouse is placed at a point (x,y) on the canvas and the left mouse button is clicked, a line segment from point (x, y) to point (50, 50) should be drawn. If one were to click at many arbitrary canvas points, your canvas would look something like this:



10. Implement a new GUI class App that combines user-defined tkinter widget classes Calculator and Mortgage displayed below:





Your GUI should look like this:

