**Chapter 11, Problem 11.17**

# Frank Mock CS 122 Test 3

# This program determines if a bank is unsafe

# The user enters a unsafe limit and if a bank's

# total assets are below this limit the bank is

# unsafe

# Get the number of banks and unsafe limit

rows **=** eval**(**input**(**"Enter the number of banks "**))**

limit **=** eval**(**input**(**"Enter the asset limit "**))**

# Create a master list to hold other lists of various sizes

masterList **=** **[]**

# The number of columns is dynamic, it depends on how many

# banks borrowed from a particular bank of row(n)

columns **=** 0

# Get bank input data from user and populate masterList

# Put each bank's data in a row

**for** i **in** range**(**rows**):**

row **=** **[]** # Add an empty new Row

columns **=** 2

balance **=** eval**(**input**(**"What is the balance of bank " **+** str**(**i**)** **+** "? "**))**

b **=** eval**(**input**(**"How many banks borrowed from bank " **+** str**(**i**)** **+** "? "**))**

columns **+=** 2**\***b

**for** j **in** range**(**columns**):**

**if** j **==** 0**:**

row**.**append**(**balance**)**

**elif** j **==** 1**:**

row**.**append**(**b**)**

**elif** j **>** 1 **and** j**%**2 **==** 0**:**

id **=** eval**(**input**(**"Enter Bank ID "**))**

row**.**append**(**id**)**

**elif** j **>** 1 **and** j**%**2 **!=** 0**:**

borrowed **=** eval**(**input**(**"Enter the amount bank borrowed "**))**

row**.**append**(**borrowed**)**

masterList**.**append**(**row**)**

# A list to hold the unsafe banks

unsafeBanks **=** **[]**

# Determine which banks are unsafe

# A bank's total assets < limit == unsafe

**for** i **in** range**(**rows**):**

assets **=** 0

bank **=** i

**for** j **in** range**(**len**(**masterList**[**i**])):**

**if** j **==** 0**:**

assets **+=** masterList**[**i**][**j**]**

**elif** j **>** 1 **and** j**%**2 **!=** 0**:**

assets **+=** masterList**[**i**][**j**]**

**else:**

**pass**

**print(**"Total assets of bank " **+** str**(**bank**)** **+** " = " **+** str**(**assets**))**

**if** assets **<** limit**:**

unsafeBanks**.**append**(**bank**)**

# Print the results to the user

numUnsafeBanks **=** len**(**unsafeBanks**)**

**print(**"There are " **+** str**(**numUnsafeBanks**)** **+** " unsafe banks."**)**

**if** numUnsafeBanks **>** 0**:**

**print(**"The unsafe banks are: "**,** end**=**""**)**

**for** i **in** unsafeBanks**:**

**print(**str**(**i**)** **+** " "**)**

**Program run and output:**

Enter the number of banks 5

Enter the asset limit 201

What is the balance of bank 0? 25

How many banks borrowed from bank 0? 2

Enter Bank ID 1

Enter the amount bank borrowed 100.5

Enter Bank ID 4

Enter the amount bank borrowed 320.5

What is the balance of bank 1? 125

How many banks borrowed from bank 1? 2

Enter Bank ID 2

Enter the amount bank borrowed 40

Enter Bank ID 3

Enter the amount bank borrowed 85

What is the balance of bank 2? 175

How many banks borrowed from bank 2? 2

Enter Bank ID 0

Enter the amount bank borrowed 125

Enter Bank ID 3

Enter the amount bank borrowed 75

What is the balance of bank 3? 75

How many banks borrowed from bank 3? 1

Enter Bank ID 0

Enter the amount bank borrowed 125

What is the balance of bank 4? 181

How many banks borrowed from bank 4? 1

Enter Bank ID 2

Enter the amount bank borrowed 125

Total assets of bank 0 = 446.0

Total assets of bank 1 = 250

Total assets of bank 2 = 375

Total assets of bank 3 = 200

Total assets of bank 4 = 306

There are 1 unsafe banks.

The unsafe banks are: 3

**Chapter 12, Problem 12.22**

# Chapter 12, Problem 12.22

# This program displays a 3x3 grid of

# CoinLabels that flip when a user

# clicks on them

**from** tkinter **import** **\***

# CoinLabel extends Label

# Button-1 is bound to the flip method

**class** **CoinLabel(**Label**):**

**def** \_\_init\_\_**(**self**,** container**,** text**):**

Label**.**\_\_init\_\_**(**self**,** container**,**

text **=** text**,** font **=** "Arial 28 bold"**)**

self**.**bind**(**"<Button-1>"**,** self**.**flip**)**

# The flip method changes the text on a CoinLabel object

# The left mouse button is bound to this method

**def** flip**(**self**,** event**):**

**if** self**[**"text"**]** **==** "H"**:**

self**[**"text"**]** **=** "T"

**else:**

self**[**"text"**]** **=** "H"

# The main class creates the GUI window and creates the 3X3

# grid of CoinLables. The application runs the root window

# loop.

**class** **MainClass:**

**def** \_\_init\_\_**(**self**):**

# Create a root window and give it a title

root **=** Tk**()**

root**.**title**(**"Flip Coins"**)**

# Frame is a child of the root window

frame **=** Frame**(**root**)**

# Place the frame in the root window

frame**.**pack**()**

# Create a 3X3 grid of CoinLables

**for** i **in** range**(**3**):**

**for** j **in** range**(**3**):**

CoinLabel**(**frame**,** text **=** "H"**).**grid**(**row **=** i**,** column **=** j**)**

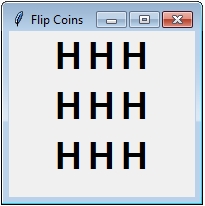
root**.**mainloop**()** # Create an event loop

# Create a MainClass object

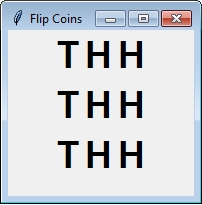
MainClass**()**

**Program run and output on next page:**

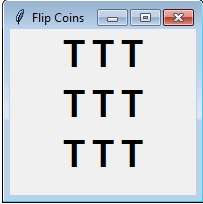
**All coins flipped to heads when the program starts**

****

**The first column with all CoinLabels flipped to tails**

****

**All coins flipped to tails**

****

**Chapter 13, Problem 13.13**

# Frank Mock CS 122 Test 3

# Chapter 13, Problem 13.13

**from** tkinter **import** **\***

**def** displayGraph**(**canvas**,** vertices**,** edges**):**

radius **=** 3

# Draw the nodes and the node numbers

**for** vertex**,** x**,** y **in** vertices**:**

canvas**.**create\_text**(**x **-** 2 **\*** radius**,** y **-** 2 **\*** radius**,**

text **=** str**(**vertex**),** tags **=** "graph"**)**

canvas**.**create\_oval**(**x **-** radius**,** y **-** radius**,** x **+** radius**,**

y **+** radius**,** fill **=** "black"**,** tags **=** "graph"**)**

# Draw each vertex

**for** v1**,** v2 **in** edges**:**

canvas**.**create\_line**(**vertices**[**v1**][**1**],** vertices**[**v1**][**2**],**

vertices**[**v2**][**1**],** vertices**[**v2**][**2**],** tags **=** "graph"**)**

# The main function gets the graph input file from the user and

# uses the displayGraph function to draw a graphic representation

# of the graph

**def** main**():**

# Prompt the user to enter a file name

f1 **=** input**(**"Enter the file for the graph: "**).**strip**()**

# open file for input

infile **=** open**(**f1**,** "r"**)**

# Read the first line from the file

numberOfVertices **=** eval**(**infile**.**readline**())**

#print(numberOfVertices)

# Create a list to hold vertices and one to hold edges

vertices **=** **[]**

edges **=** **[]**

# Fill the vertices and edges list with data from the graph file

**for** i **in** range**(**numberOfVertices**):**

# Read the info for one vertex

items **=** infile**.**readline**().**strip**().**split**()**

vertices**.**append**([**eval**(**items**[**0**]),** eval**(**items**[**1**]),** eval**(**items**[**2**])])**

**for** j **in** range**(**3**,** len**(**items**)):**

edges**.**append**([**eval**(**items**[**0**]),** eval**(**items**[**j**])])**

#print(vertices)

#print(edges)

# Create a window

window **=** Tk**()**

window**.**title**(**"Display Graph"**)**

# Create a frame to hold a canvas object

frame1 **=** Frame**(**window**)**

frame1**.**pack**()**

# The canvas object will have text, ovals and lines drawn on it

# by the display graph function

canvas **=** Canvas**(**frame1**,** width **=** 300**,** height **=** 200**)**

canvas**.**pack**()**

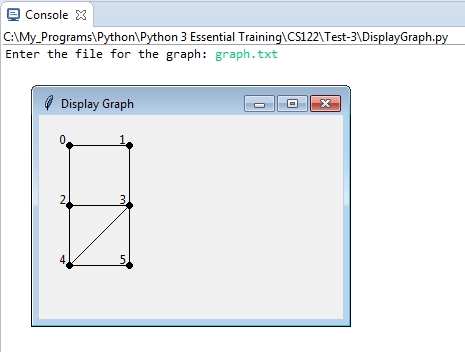
# Call the displayGraph function to draw the graph

displayGraph**(**canvas**,** vertices**,** edges**)**

# The main loop

window**.**mainloop**()**

main**()**

****

**The contents of graph.txt**

6

0 30 30 1 2

1 90 30 0 3

2 30 90 0 3 4

3 90 90 1 2 4 5

4 30 150 2 3 5

5 90 150 3 4

**Chapter 14, Problem 14.2**

# This program determines which numbers occur the

# from a given list of numbers

**def** main**():**

**print(**"This program determines which numbers occur the most "**)**

**print(**"frequently in a list of numbers"**)**

**print(**""**)**

# Main loop of program

**while** **True:**

s **=** input**(**"Enter numbers, each separated by a space: "**).**strip**()**

numbers **=** **[**eval**(**x**)** **for** x **in** s**.**split**()]**

# Create an empty dictionary

dictionary **=** **{}**

# Count the number occurences and store values in

# dictionary

**for** number **in** numbers**:**

**if** number **in** dictionary**:**

dictionary**[**number**]** **+=** 1

**else:**

dictionary**[**number**]** **=** 1

# Get the maximum values

maxCount **=** max**(**dictionary**.**values**())**

pairs **=** list**(**dictionary**.**items**())**

# Reverse pairs in the list

items **=** **[[**x**,** y**]** **for** **(**x**,** y**)** **in** pairs**]**

# Display numbers to the user

**print(**"The numbers that occur the most are "**,** end **=** ""**)**

**for** **(**x**,** y**)** **in** items**:**

**if** y **==** maxCount**:**

**print(**x**,** end **=** " "**)**

# Ask the user if they would like to play again

again **=** input**(**"Again? (y/n)"**)**

**if** again **==** 'n'**:**

**print(**"Goodbye"**)**

**break**

main**()**

**Program run and output on the next page**

This program determines which numbers occur the most

frequently in a list of numbers

Enter numbers, each separated by a space: 16 22 5 4 8 60 9 5 22 40 75

The numbers with the most occurrence are 5 22 Again? (y/n)y

Enter numbers, each separated by a space: 12 8 7 99 32 16 2 80 2 35 19

The numbers with the most occurrence are 2 Again? (y/n)n

Goodbye

**Chapter 14, Problem 14.4**

**from** tkinter **import** **\***

**import** tkinter**.**messagebox

# Needed to allow user to browse and select a file

**from** tkinter**.**filedialog **import** askopenfilename

# This program will determine which letters occur the

# most frequently in text file. The program lets the

# user select the text file by using the tkinter

# askopenfilename dialog.

# Gets the file name and passes it to the analyzeFile

# function

**def** showResult**():**

analyzeFile**(**filename**.**get**())**

# This function opens a file for reading and then analyzes

# the contents by counting the occurences of each letter

**def** analyzeFile**(**filename**):**

**try:**

infile **=** open**(**filename**,** "r"**)** # Open the file

counts **=** 26 **\*** **[**0**]** # Create and initialize counts

**for** line **in** infile**:**

# Invoke the countLetters function to count each letter

countLetters**(**line**.**lower**(),** counts**)**

# Display results

**for** i **in** range**(**len**(**counts**)):**

**if** counts**[**i**]** **!=** 0**:**

text**.**insert**(**END**,** chr**(**ord**(**'a'**)** **+** i**)** **+**

" appears " **+** str**(**counts**[**i**])**

**+** **(**" time" **if** counts**[**i**]** **==** 1 **else** " times"**)** **+**

"\n"**)**

infile**.**close**()** # Close file

**except** IOError**:**

tkinter**.**messagebox**.**showwarning**(**"Analyze File"**,**

"File " **+** filename

**+** " does not exist"**)**

# Count each letter in the string

**def** countLetters**(**line**,** counts**):**

**for** ch **in** line**:**

**if** ch**.**isalpha**():**

counts**[**ord**(**ch**)** **-** ord**(**'a'**)]** **+=** 1

# Presents the user with a file dialog so they may

# choose the file to be analyzed

**def** openFile**():**

filenameforReading **=** askopenfilename**()**

filename**.**set**(**filenameforReading**)**

######## Begin creating the GUI #########

# Create a window and give it a title

window **=** Tk**()**

window**.**title**(**"Occurrence of Letters"**)**

# Create a frame to hold main window and scrollbar

f1 **=** Frame**(**window**)**

f1**.**pack**()**

# Create scrollbar and place inside the frame, f1

scrollbar **=** Scrollbar**(**f1**)**

scrollbar**.**pack**(**side **=** RIGHT**,** fill **=** Y**)**

text **=** Text**(**f1**,** width **=** 40**,** height **=** 10**,** wrap **=** WORD**,**

yscrollcommand **=** scrollbar**.**set**)**

text**.**pack**()**

scrollbar**.**config**(**command **=** text**.**yview**)**

# Create another frame to hold labels and buttons

f2 **=** Frame**(**window**)**

f2**.**pack**()**

Label**(**f2**,** text **=** "Enter a filename: "**).**pack**(**side **=** LEFT**)**

filename **=** StringVar**()**

Entry**(**f2**,** width **=** 20**,** textvariable **=** filename**).**pack**(**side **=** LEFT**)**

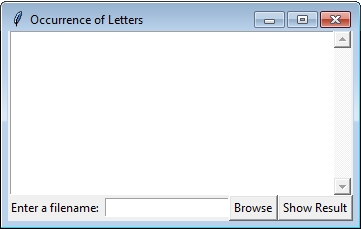
Button**(**f2**,** text **=** "Browse"**,** command **=** openFile**).**pack**(**side **=** LEFT**)**

Button**(**f2**,** text **=** "Show Result"**,**

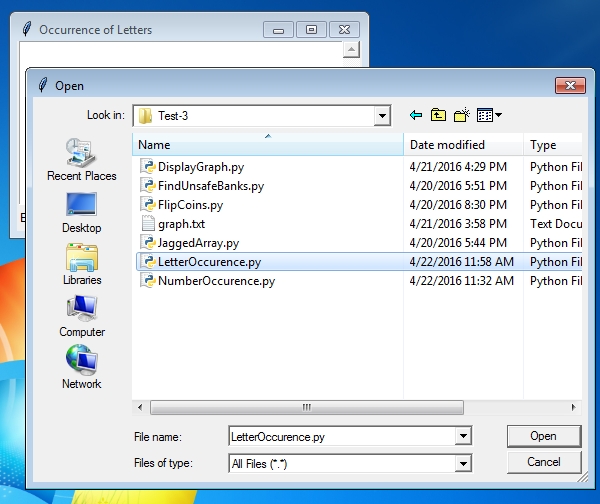
command **=** showResult**).**pack**(**side **=** LEFT**)**

# Create an event loop

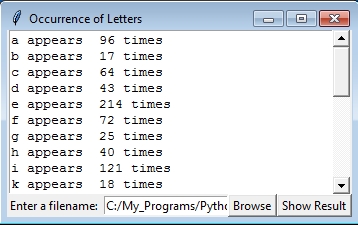
window**.**mainloop**()**



**Allows user to browse and select the file to be analyzed**



**The results of the analysis are displayed to the user**

****

**Chapter 14, Problem 14.7**

**from** tkinter **import** **\***

**import** tkinter**.**messagebox

# Class used to display the bar chart

**from** BarChart **import** BarChart

# Needed to allow user to browse and select a file

**from** tkinter**.**filedialog **import** askopenfilename

# This program will determine which letters occur the

# most frequently in text file. The program lets the

# user select the text file by using the tkinter

# askopenfilename dialog. The program will display the

# results in text form and as a bar chart

# Gets the file name and passes it to the analyzeFile

# function

**def** showResult**():**

analyzeFile**(**filename**.**get**())**

# This function opens a file for reading and then analyzes

# the contents by counting the occurences of each letter

**def** analyzeFile**(**filename**):**

**try:**

infile **=** open**(**filename**,** "r"**)** # Open the file

counts **=** 26 **\*** **[**0**]** # Create and initialize counts

**for** line **in** infile**:**

# Invoke the countLetters function to count each letter

countLetters**(**line**.**lower**(),** counts**)**

# A list to hold the data that will be represented in

# the bar chart

data **=** **[]**

# Build the data multidimensional list that

# the BarChart object will use

**for** i **in** range**(**len**(**counts**)):**

s **=** chr**(**ord**(**'a'**)** **+** i**)**

data**.**append**([**counts**[**i**]** **+** 8**,** s**,** "red"**])**

# Create a BarChart object and pack it in to the GUI window

# This will display the bar chart in the window

c **=** BarChart**(**window**,** data**)**

c**.**pack**()**

# Display textual number occurrence results

**for** i **in** range**(**len**(**counts**)):**

**if** counts**[**i**]** **!=** 0**:**

text**.**insert**(**END**,** chr**(**ord**(**'a'**)** **+** i**)** **+**

" appears " **+** str**(**counts**[**i**])**

**+** **(**" time" **if** counts**[**i**]** **==** 1 **else** " times"**)** **+**

"\n"**)**

# Close file

infile**.**close**()**

**except** IOError**:**

tkinter**.**messagebox**.**showwarning**(**"Analyze File"**,**

"File " **+** filename

**+** " does not exist"**)**

# Count each letter in the string

**def** countLetters**(**line**,** counts**):**

**for** ch **in** line**:**

**if** ch**.**isalpha**():**

counts**[**ord**(**ch**)** **-** ord**(**'a'**)]** **+=** 1

# Presents the user with a file dialog so they may

# choose the file to be analyzed

**def** openFile**():**

filenameforReading **=** askopenfilename**()**

filename**.**set**(**filenameforReading**)**

######## Begin creating the GUI #########

# Create a window and give it a title

window **=** Tk**()**

window**.**title**(**"Occurrence of Letters"**)**

f1 **=** Frame**(**window**)**

f1**.**pack**()**

scrollbar **=** Scrollbar**(**f1**)**

scrollbar**.**pack**(**side **=** RIGHT**,** fill **=** Y**)**

text **=** Text**(**f1**,** width **=** 40**,** height **=** 10**,** wrap **=** WORD**,**

yscrollcommand **=** scrollbar**.**set**)**

text**.**pack**()**

scrollbar**.**config**(**command **=** text**.**yview**)**

f2 **=** Frame**(**window**)**

f2**.**pack**()**

Label**(**f2**,** text **=** "Enter a filename: "**).**pack**(**side **=** LEFT**)**

filename **=** StringVar**()**

Entry**(**f2**,** width **=** 20**,** textvariable **=** filename**).**pack**(**side **=** LEFT**)**

Button**(**f2**,** text **=** "Browse"**,** command **=** openFile**).**pack**(**side **=** LEFT**)**

Button**(**f2**,** text **=** "Show Result"**,**

command **=** showResult**).**pack**(**side **=** LEFT**)**

# Create an event loop

window**.**mainloop**()**

