CS 2316

Individual Homework 6 –Arithmetic Number Search Puzzle

Due: Wednesday, March 2nd, before 11:55 PM

Out of 100 points

File to submit: HW6.py

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use Piazza

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- Do not wait until the last minute to do this assignment in case you run into problems.
- **Read the entire specifications document before starting this assignment.**

Premise

In this homework you will be creating a program to solve arithmetic number search puzzles. You can assume that each arithmetic expression will consist of two digits (between 0 and 9 inclusive) and one basic mathematical operation.

File Format Information:
You will be required to read in two CSV files.

The first CSV file will consist of rows of integers and the Python operators for addition (+), subtraction (-), multiplication (*), and division (/). Each row corresponds to a row in the number search puzzle, and you can assume that each row will have the same number of elements.

Here is an example of what a row in the file could be:

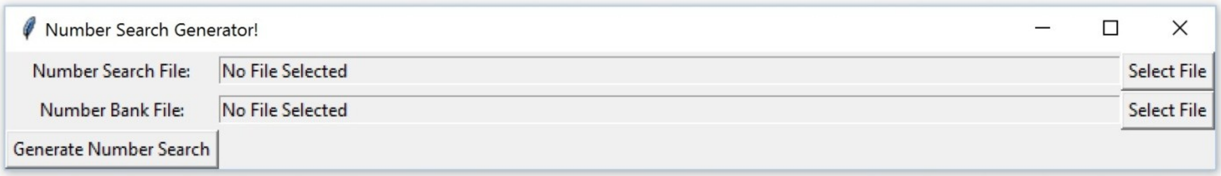
5, +, 4, 9, /, 8, *, 2, -

The next line in the file would have the same number of elements.

The second CSV file will simply contain one column of the evaluated integers that will be searched for in the puzzle.

Method Name: __init__

This method is automatically called whenever you create a new instance of your GUI class. The __init__ method is responsible for arranging and initializing your GUI. You should create a GUI that looks like the picture below.



You may generate this window however you wish, but it is highly suggested that you use the grid layout manager. Things to note about the GUI you see:

- Consider putting all of the widgets in a frame because you will be using multiple frames throughout this homework.
- The entry box between the "Number Search File:" label and the "Select File" button and the entry box between the "Number Bank File:" label and the "Select File" button each must have a width of 100, must be readonly, and must initially contain some message (e.g., "No File Selected") to indicate no file has been selected.
    - On the line with the "Number Search File:" label, the "Select File" button calls the method called openNSClicked
    - On the line with the "Number Bank File:" label, the "Select File" button calls the method called openNBClicked
    - The "Generate Number Search" button calls the method generate
- The title of the GUI is "Number Search Generator!"

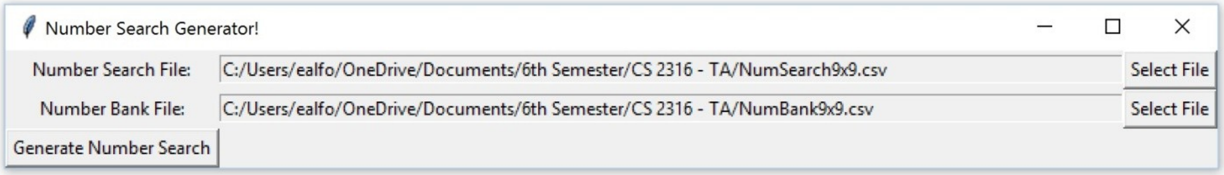
Method Name: openNSClicked

This method will simply launch a file open dialog box and prompt the user to select a number search file. The function used to accomplish this is **filedialog.askopenfilename()**. After the user selects a file, the text for the Entry box beside "Number Search File:" label will be set to the file path. **It is suggested that you store the file name returned in the "filename" object variable, so that other methods can access the string using "self.fileName".**

MethodName: openNBClicked

Similar to openNSClicked, this method will launch a file open dialog box and prompt the user to select a number bank file. After the user selects a file, the text for the Entry box beside "Number Bank File:" label will be set to the file path. Store the file name returned in the "filename1" object variable, so that other methods can access the string using "self.fileName1".

After both openNSClicked and openNBClicked have been called the GUI should look like the picture below.

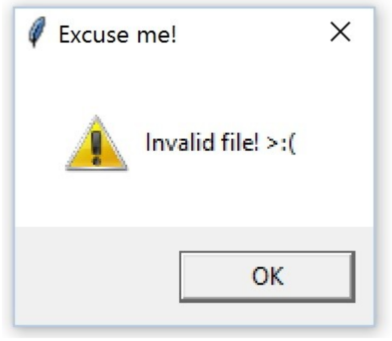


C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumSearch9x9.csv

C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumBank9x9.csv

Method Name: readFiles

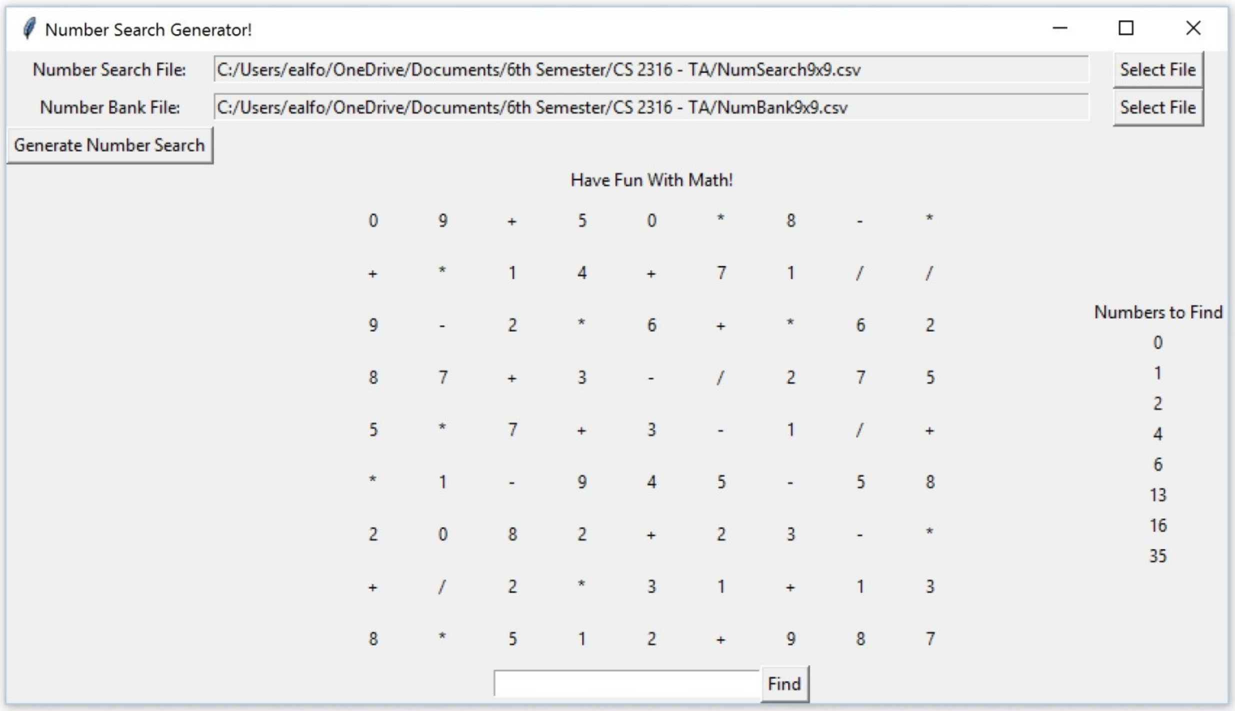
This method should access the names of the two files, and read them. It should check that all the characters are valid (i.e., integers or one the four operators +,=,*,/). If the files do not meet that criteria then it can't be read and there should be an error message with a title and message indicating that the file could not be read (see example messagebox below). When reading in the CSV file, all whitespace should be stripped. The puzzle should be fixed if white space needs to be stripped. This method should make a list of each line in the CSV file nested in a list(e.g. [['5', '+', '4', '9', '/', '8', '*', '2', '-'],[next row in csv file],etc.) and the list should be an instance variable (i.e., self.numSearchLines).



Invalid file! >:(

Method Name: generate

This method should take the instance variable from the readFile method and put it into the format of the picture below. The "Find" button's command is tied to the find method (described below).



Number Search File: C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumSearch9x9.csv

Number Bank File: C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumBank9x9.csv

Have Fun With Math!

0	9	+	5	0	*	8	-	*
+	*	1	4	+	7	1	/	/
9	-	2	*	6	+	*	6	2
8	7	+	3	-	/	2	7	5
5	*	7	+	3	-	1	/	+
*	1	-	9	4	5	-	5	8
2	0	8	2	+	2	3	-	*
+	/	2	*	3	1	+	1	3
8	*	5	1	2	+	9	8	7

Numbers to Find
0
1
2
4
6
13
16
35

It is highly suggested that you use the grid layout manager! Other things to note about the GUI:

- Your Number Search puzzle should be labeled (e.g., "Have Fun With Math!")
- Each number or symbol is its own label and between each number or symbol there is enough space so that you can visually separate them.
- It is suggested to put the numbers and symbols in a frame.
- To the right is a number bank of a list of evaluated integers to search and solve for. This can be put into another frame, and should also be titled.

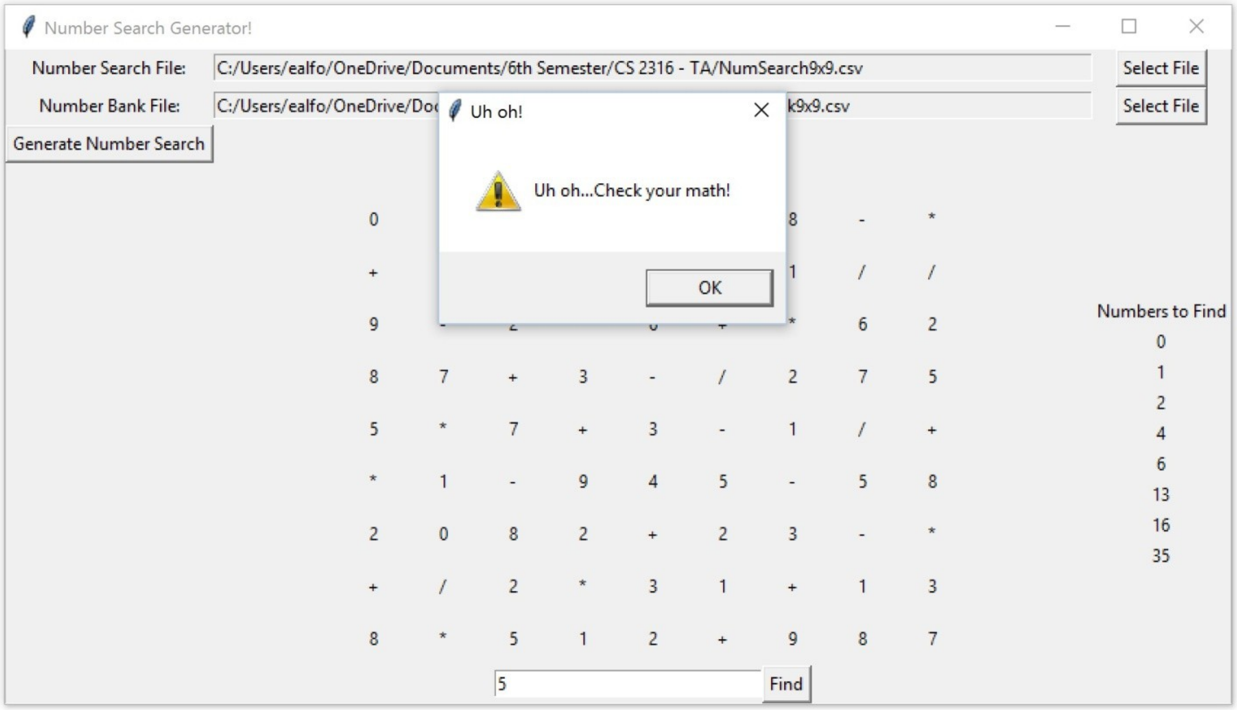
- Underneath the number search is an entry box with width 30 and a button “Find” which can be put into another frame.
- The text in the entry box next to the “Find” button should be able to be accessed in other methods.

Make sure to store a list as an instance variable called self.numbersToFind that contains the label objects created when you make the number bank. Then make sure to store another list as an instance variable called self.numbers that contains the label objects of the numbers and operators in the number search puzzle. Make sure the indices MATCH UP! If the number or symbol is at (0,0) like in the picture above “0” then its label object should be at index 0.

EX.
self.numSearchLines=[…,['5', '+', '4', '9', '/', '8', '*', '2', '-'], ['+', '7', '2', '5', '/', '0', '*', '1'],…]
self.numbersToFind=[pointer to Label(self.rootWin, text=”NONE”), etc. for number bank]
self.numbers=[[pointer to Label(self.rootWin, text = “0”), etc for row 1],etc.]
Notice that self.numbers is a 2D list

Method Name: findStartingCoords
In this method you are going to return a list of all the coordinates of the first number of the expression entered in the entry box next to the “Find” button. The coordinates should be in a tuple in the list. For the expression “8/2” the findStartingCoords method would return [(0,6), (3,0), (5,8), (6,2), (8, 0), (8,7)] because there is an “8” at those positions.

Method Name: find
Get the expression in the entry box next to the “Find” button and check if the evaluated answer is in the number bank. The eval() function may be useful for this task. If the expression typed into the entry box is not valid, is not in the puzzle, or is does not evaluate to a number in the number bank you should indicate to the user that the input was not found. (In the example the message is “Uh oh… Check your math!”)


```
Number Search Generator!
Number Search File:  C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumSearch9x9.csv   Select File
Number Bank File:    C:/Users/ealfo/OneDrive/Doc                    k9x9.csv                          Select File
Generate Number Search

Uh oh!
Uh oh...Check your math!
OK

0                          8    -    *
+                          1    /    /
9    -    2    0    +      *    6    2                Numbers to Find
8    7    +    3    -    /    2    7    5                  0
5    *    7    +    3    -    1    /    +                  1
*    1    -    9    4    5    -    5    8                  2
2    0    8    2    +    2    3    -    *                  4
+    /    2    *    3    1    +    1    3                  6
8    *    5    1    2    +    9    8    7                  13
                                                           16
                                                           35
          5                  Find
```

In this method you are going to get the returned list from findStartingCoords. Iterate through findStartingCoords and for each findStartingCoords you should search in each different direction

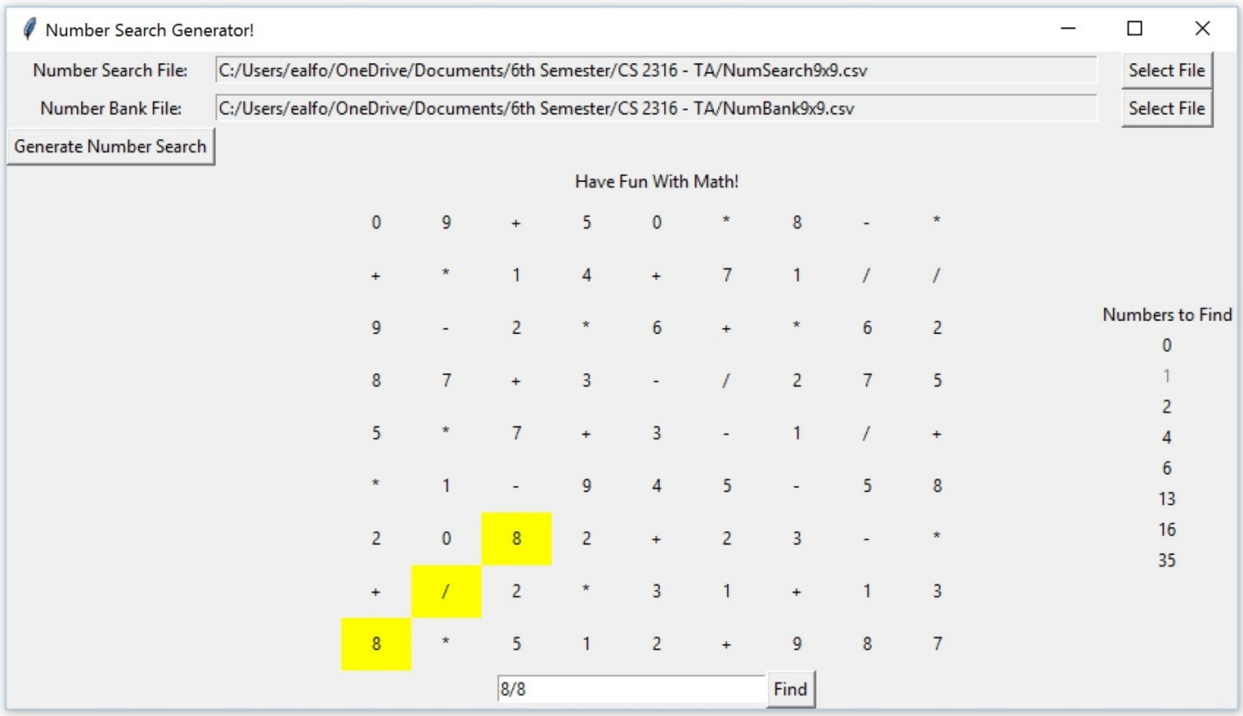
Different Directions:

- Left to Right match
- Right to Left match
- Down Right match
- Down Left match
- Up Right match
- Up Left match
- Up match
- Down match

For each direction, make sure your indices are not out of bounds (you can use try and except). When searching along in a direction, if the next number or symbol you see is not the next number or symbol in the expression from the entry box then you should raise a value error or continue. This will cause your code to stop trying in that direction. If you find the next number or symbol is the next matching number or symbol in the expression then make sure you save those coordinates. Once you find the coordinates for the entire expression you should return it in a variable designated coordList.

Method Name: updateGUI

If the expression evaluates to a number in the number search then each number or symbol in the expression should be changed to yellow. Use your saved labels in self.numbersToFind and self.numbers to change the color of the numbers and operators in the number search puzzle and the number bank. You should iterate through the pointer to the Label and configure them to the appropriate colors.



Number Search Generator!

Number Search File: C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumSearch9x9.csv

Number Bank File: C:/Users/ealfo/OneDrive/Documents/6th Semester/CS 2316 - TA/NumBank9x9.csv

Have Fun With Math!

0	9	+	5	0	*	8	-	*
+	*	1	4	+	7	1	/	/
9	-	2	*	6	+	*	6	2
8	7	+	3	-	/	2	7	5
5	*	7	+	3	-	1	/	+
*	1	-	9	4	5	-	5	8
2	0	8	2	+	2	3	-	*
+	/	2	*	3	1	+	1	3
8	*	5	1	2	+	9	8	7

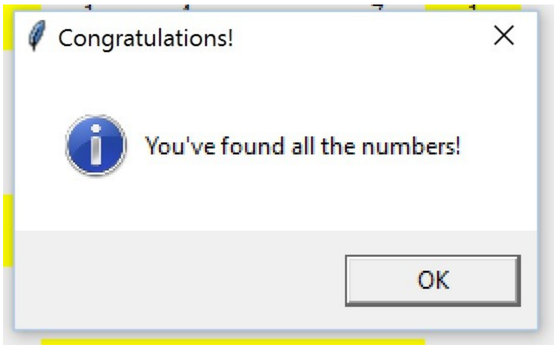
Numbers to Find
0
1
2
4
6
13
16
35

8/8

Things to note about the GUI above:

- When you find a correct expression in the number search puzzle, the number in the number bank changes to grey. (You can use .config(fg=”grey”) to accomplish this.)
- When you find an expression in the number search, the background of each number/symbol in the found expression should change to yellow. (You can use .config(bg=”yellow”) to accomplish this)

When all of the numbers in the number bank are found, indicate to the user that they have solved the puzzle in a messagebox and destroy the game window.


Congratulations!

You've found all the numbers!

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications.

The GUI		**10**
The GUI looks like it should	10	
openNSClicked / openNBClicked		**10**
Replaces the Number Search Entry box text with file path	5	
Replaces the Number Bank Entry box text with file path	5	
readFiles		**10**
Properly reads in both csv files	5	
Makes data accessible in other files	5	
generate		**30**
Makes number bank	10	
Makes the number search puzzle	5	
Makes the find Entry box	5	
Makes self.numbersToFind correctly	5	
Makes self.numbers correctly	5	
findStartingCoords		**10**
findStartingCoords is correct	5	

returns findStartingCoords	5	
find		**18**
Left to Right match	2	
Right to Left match	2	
Down Right match	2	
Down Left match	2	
Up Right match	2	
Up Left match	2	
Up match	2	
Down match	2	
Handles if the expression is not valid	2	
updateGUIs		**12**
Changes the color of the number/symbol in the number search	5	
Correctly changes the color of the number in the number bank	5	
Indicates to user when the puzzle is solved and exits correctly	2	