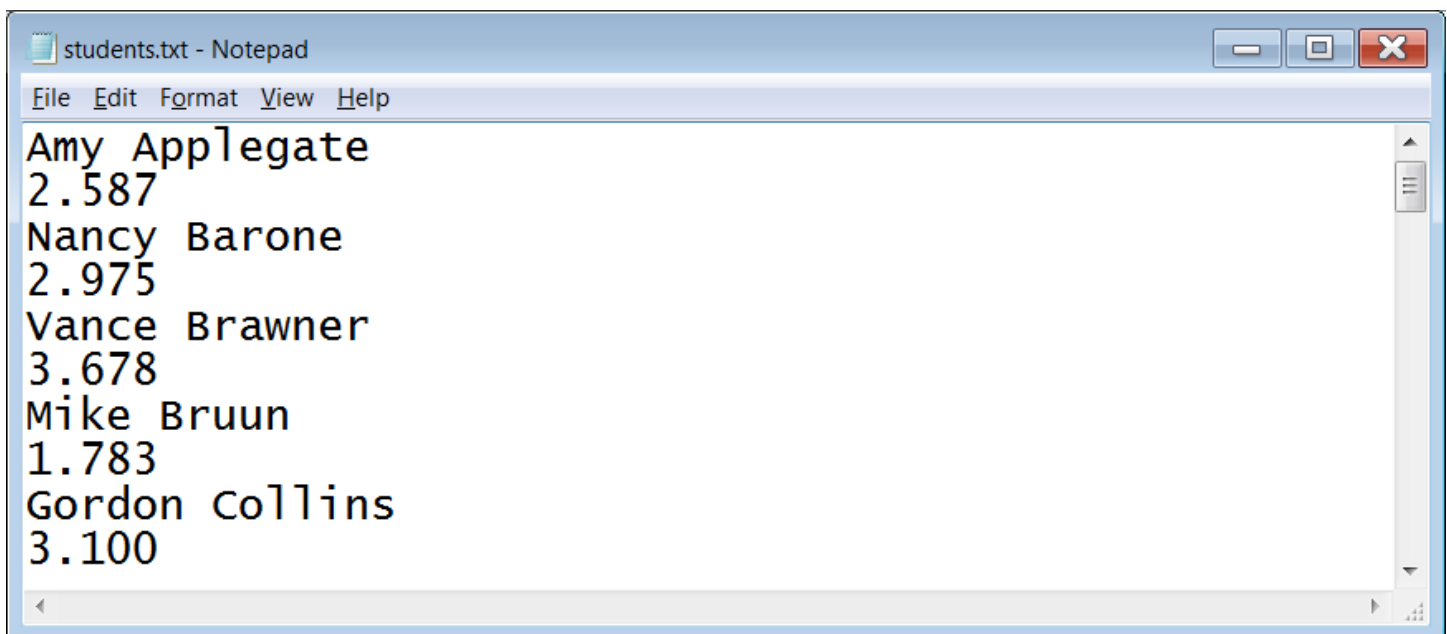


Computer Science 1-Honors	Lab 17A Practice/Perform Major Python Assignment
Manipulating Text Files of Students	60, 70, 80, 90, 100 & 110 Point Versions
Assignment Purpose: The purpose of this program is to demonstrate knowledge of reading in from a text file, interpreting the information, and writing to one or more other text files.	

Write a program that reads the data from a text file called "**students.txt**" and then manipulates the data in various ways. The file is organized to store 2 different pieces of student information. Specifically, the file stores the **name** and **gpa** of each student. Each piece of information is on a separate line in the file. The first five records of the text file are shown below.



```

students.txt - Notepad
File Edit Format View Help
Amy Applegate
2.587
Nancy Barone
2.975
Vance Brawner
3.678
Mike Bruun
1.783
Gordon Collins
3.100
  
```

Keep in mind that since "**students.txt**" is a *text* file, all of the information is *stored* and *read in* as *strings*. While this is fine for the **name**, it is no good for the **gpa**. You need to convert the **gpa** string to a real number with **float**.

Lab 17A Student Version	Do not copy this file, which is provided.
<pre> 1 # Lab17Ast.py 2 # "Manipulating Text Files" 3 # This is the student, starting version of Lab 17A. 4 # At minimum, students need to implement the 5 # <readData> and <displayData> procedures. 6 </pre>	

```

7
8 def heading():
9     print()
10    print("*****")
11    print("Lab 17A, Manipulating Text Files")
12    print("60 Point Version")
13    print("By: JOHN SMITH") # Substitute your own name here.
14    print("*****")
15    print("\n")
16
17
18 def readData(studentNames,studentGPAs):
19     pass
20
21
22 def displayData(studentNames,studentGPAs):
23     pass
24
25
26 def computeAndDisplayAverageGPA(studentGPAs):
27     pass
28
29
30 def capitalizeNames(studentNames):
31     pass
32
33
34 def writeData(studentNames,studentGPAs):
35     pass
36
37
38 def sortDataByGPA(studentNames,studentGPAs):
39     pass
40
41
42
43 #####
44 #  MAIN  #
45 #####
46
47 heading()
48 studentNames = []
49 studentGPAs = []
50 readData(studentNames,studentGPAs)      # Do for 60 point version
51 capitalizeNames(studentNames)           # Do for 80 point version
52 sortDataByGPA(studentNames,studentGPAs) # Do for 110 point version
53 displayData(studentNames,studentGPAs)   # Do for 60 point version
54 computeAndDisplayAverageGPA(studentGPAs) # Do for 70 point version
55 writeData(studentNames,studentGPAs)     # Do for 90/100 point version

```

60 Point Version Specifics

For the 60-point version you need to implement procedures **readData** and **displayData**. The **readData** procedure needs to read in all of the information from the "**students.txt**" file and store it in the 2 parallel arrays: **studentNames** and **studentsGPAs**. The **displayData** procedure displays the data from these arrays in an organized manner, distinguishing the student names from their GPAs and skipping a line after each student record. (HINT: Refer to program example **TextFiles18.py**.)

60-Point Version *Monitor* Output

Since the output is so long, only the first 3 and the last 3 student records are shown below.

```
----jGRASP exec: python Lab17Av60.py
```

```
*****  
Lab 17A, Manipulating Text Files  
60 Point Version  
By: JOHN SMITH  
*****
```

```
Student Name: Amy Applegate  
Student GPA: 2.587
```

```
Student Name: Nancy Barone  
Student GPA: 2.975
```

```
Student Name: Vance Brawner  
Student GPA: 3.678
```

```
: : : :
```

```
Student Name: Michael Ward  
Student GPA: 3.451
```

```
Student Name: Cheryl Willis  
Student GPA: 3.576
```

```
Student Name: Ziggy Zighlander  
Student GPA: 3.777
```

```
----jGRASP: operation complete.
```

70 Point Version Specifics

For the 70-point version you need to compute the average of all of the student GPAs and display it at the end of the program output, along with the student count. This means you are now required to convert the GPAs to real numbers (using the **float** command) when you read them in from the file. Keep in mind that the average needs to be rounded to 2 decimal places. Computing the average can either be done by implementing the **computeAndDisplayAverageGPA** procedure (heading is provided) or it can be done in the **displayData** procedure as the student information is being displayed, similar to what is done in program example **TextFiles18.py**.

70-Point Version *Monitor* Output

```
----jGRASP exec: python Lab17Av70.py
```

```
*****  
Lab 17A, Manipulating Text Files  
70 Point Version  
By: JOHN SMITH  
*****
```

```
Student Name: Amy Applegate  
Student GPA: 2.587
```

```
Student Name: Nancy Barone  
Student GPA: 2.975
```

```
Student Name: Vance Brawner  
Student GPA: 3.678
```

```
:      :      :      :
```

```
Student Name: Michael Ward  
Student GPA: 3.451
```

```
Student Name: Cheryl Willis  
Student GPA: 3.576
```

```
Student Name: Ziggy Zighlander  
Student GPA: 3.777
```

```
# of Students: 36  
Average GPA: 2.97
```

NOTE: 5 points will be deducted if the **Average GPA is not rounded to 2 decimal places.**

```
----jGRASP: operation complete.
```

80 Point Version Specifics

For the 80-point version you need to CAPITALIZE all of the letters in all of the student names. You can do this by implementing the **capitalizeNames** procedure (heading is provided) or it can be done as you are reading in the names in the **readData** procedure.

80-Point Version *Monitor* Output

```
----jGRASP exec: python Lab17Av80.py

*****
Lab 17A, Manipulating Text Files
80 Point Version
By: JOHN SMITH
*****

Student Name: AMY APPLGATE
Student GPA:  2.587

Student Name: NANCY BARONE
Student GPA:  2.975

Student Name: VANCE BRAUNER
Student GPA:  3.789

:      :      :      :

Student Name: MICHAEL WARD
Student GPA:  3.451

Student Name: CHERYL WILLIS
Student GPA:  3.576

Student Name: ZIGGY ZIGHLANDER
Student GPA:  3.777

# of Students: 36
Average GPA: 2.97

----jGRASP: operation complete.
```

90 Point Version Specifics

For the 90-point version you need to implement the **writeData** procedure. Nothing new will be displayed on the screen, but a new file needs to be created called "**CAPS.TXT**" which will store all of the newly CAPITALIZED student names along with their corresponding GPAs.

90-Point Version File Output

NOTE: The *monitor* output is identical to the 80-point version.

This is one continuous file shown in 3 columns:



AMY APPEGATE	BART REAGOR	LORI SALBURY
2.587	2.867	3.455
NANCY BARONE	KRISTYN RECKNER	BARBARA SALSA
2.975	3.225	3.0
VANCE BRAWNER	PAUL REIMAN	ANN SEABORN
3.789	4.0	1.785
MIKE BRUUN	ANDY REITINGER	DIANE SIMCOX
1.783	3.525	2.175
GORDON COLLINS	WILLIAM REYNOLDS	KEVIN SIMS
3.125	3.375	3.235
LAURA COLLINS	MICHELLE RITTER	PATTI SKINNER
3.999	4.0	2.743
MALI COZART	WILLIAM ROBBINS	TRACY SPRINGER
2.345	2.875	2.345
SUSAN CRAFT	BRIAN ROBERTS	CHRIS STARK
3.754	3.175	4.0
TODD DEANS	LAURI ROBERTSON	TOM TOOCH
1.23	2.925	1.456
STEVEN JOHNSON	RICHARDSON ROBINSON	MICHAEL WARD
0.785	3.21	3.451
DEBBIE MOZART	DIANA ROCKEL	CHERYL WILLIS
2.999	3.456	3.576
MARY PRIDGEN	LAURIE ROSEMBERG	ZIGGY ZIGHLANDER
1.958	4.0	3.777

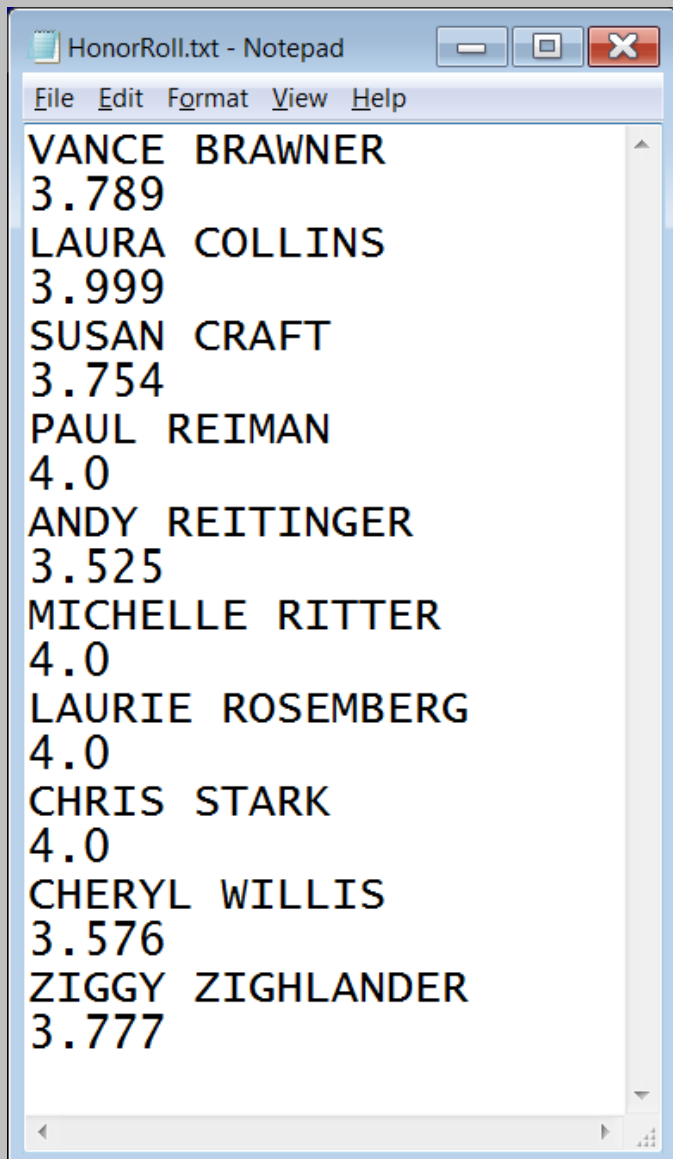
100 Point Version Specifics

For the 100-point version you need to implement the **writeData** procedure in a slightly more complicated manner. Instead of creating the **"CAPS.TXT"** file from the 90-point version, your program needs to create a file called **"HonorRoll.txt"**. Aside from the filename, the main difference between the 2 files is that **"CAPS.TXT"** stores all of the student records, while **"HonorRoll.txt"** only stores the records of students whose GPAs are above **3.5**.

100-Point Version File Output

NOTE: The *monitor* output is identical to the 80-point version.

This file is much shorter than "CAPS.TXT".



```
HonorRoll.txt - Notepad
File Edit Format View Help
VANCE BRAWNER
3.789
LAURA COLLINS
3.999
SUSAN CRAFT
3.754
PAUL REIMAN
4.0
ANDY REITINGER
3.525
MICHELLE RITTER
4.0
LAURIE ROSEMBERG
4.0
CHRIS STARK
4.0
CHERYL WILLIS
3.576
ZIGGY ZIGHLANDER
3.777
```

110 Point Version Specifics

For the 110-point version you need to implement the **sortDatabyGPA** procedure and sort the student data by GPA in *descending* (largest to smallest) order. This is not just a matter of typing the code for the *Bubble Sort* because you need to sort *parallel arrays*. If you simply perform a *Bubble Sort* on the **studentsGPA** array, that array will be sorted, but the **studentsNames** array will still be in its original order – which means the names no longer match the GPAs – which is not good. What you have to remember is that when you swap the GPAs from the **studentGPAs** array, you must also swap the corresponding names from the **studentNames** array.

NOTE: The *monitor* output and *file* output are both unique for the 110-point version.

110-Point Version *Monitor* Output

```
----jGRASP exec: python Lab17Av110.py
```

```
*****
Lab 17A, Manipulating Text Files
110 Point Version
By: JOHN SMITH
*****
```

```
Student Name: PAUL REIMAN
Student GPA: 4.0
```

```
Student Name: MICHELLE RITTER
Student GPA: 4.0
```

```
Student Name: LAURIE ROSEMBERG
Student GPA: 4.0
```

```
Student Name: CHRIS STARK
Student GPA: 4.0
```

```
Student Name: LAURA COLLINS
Student GPA: 3.999
```

```
Student Name: VANCE BRAWNER
Student GPA: 3.789
```

```
Student Name: ZIGGY ZIGHLANDER
Student GPA: 3.777
```


: : : :

Student Name: BART REAGOR
Student GPA: 2.867

Student Name: PATTI SKINNER
Student GPA: 2.743

Student Name: AMY APPELEGATE
Student GPA: 2.587

Student Name: MALI COZART
Student GPA: 2.345

Student Name: TRACY SPRINGER
Student GPA: 2.345

Student Name: DIANE SIMCOX
Student GPA: 2.175

Student Name: MARY PRIDGEN
Student GPA: 1.958

Student Name: ANN SEABORN
Student GPA: 1.785

Student Name: MIKE BRUUN
Student GPA: 1.783

Student Name: TOM TOOCH
Student GPA: 1.456

Student Name: TODD DEANS
Student GPA: 1.23

Student Name: STEVEN JOHNSON
Student GPA: 0.785

of Students: 36
Average GPA: 2.97

----jGRASP: operation complete.

110-Point Version File Output

The honor students in this file will also be sorted by GPA.



A screenshot of a Notepad window titled "HonorRoll.txt - Notepad". The window displays a list of honor students and their GPAs, sorted by GPA. The text is as follows:

Student Name	GPA
PAUL REIMAN	4.0
MICHELLE RITTER	4.0
LAURIE ROSEMBERG	4.0
CHRIS STARK	4.0
LAURA COLLINS	3.999
VANCE BRAWNER	3.789
ZIGGY ZIGHLANDER	3.777
SUSAN CRAFT	3.754
CHERYL WILLIS	3.576
ANDY REITINGER	3.525