

CSE 101 - Introduction to Programming
Assignment 2
February 2022

GENERAL INSTRUCTIONS

1. In this assignment, you need to write the code for every problem in different '**.py**' files.
2. You are only allowed to use the basic modules available in Python. You cannot use any external modules, libraries, or APIs.
3. You will create a document, citing all the online and offline resources, that you have used, in completing this assignment.
4. Your code will be checked for plagiarism against the code of your classmates as well as some sample codes available online. The institution's policies will strictly apply.
5. This time, questions will be added in phases, i.e, further questions would be added. You will be notified regarding this.
6. Start early. Resolve all your doubts with the TAs, *2 days* before the deadline.

SUBMISSION INSTRUCTIONS

1. Your code files must be renamed as A2_2021xxx_{Problem_no}.py. For example, A2_2021001_1.py, would be the python file for problem 1 for roll no. 2021001.
2. Your document with references to resources used must be in pdf format and named A2_Resources_2021xxx.pdf
3. All these files (code files and the reference pdf file) must be put in a zipped folder named A2_2021xxx.zip
4. The zip file containing everything relevant to this assignment must be submitted before the deadline on google classroom.

PROBLEMS

Q 1. Word and Words!!

For this problem, you have to write a program that performs some operations on a text file. The program offers the user a choice to select one operation at a time. The operations are listed below:

1. **Display specific Word Count:** The user is prompted to input the word whose count is required. If the word exists in the text file, print its count and if not then print 'Word does not exist'.
2. **Display Unique Words:** For this choice, you just print all the unique words present in the text. (Note that a word is any sequence of characters/digits/alphabets enclosed between a space)
3. **Display all Word Counts:** For this option, you have to print all the unique words and their corresponding word counts. (*Hint: Use a dictionary!*)
4. **Replace Word:** You prompt the user to input two words, say Word1 and Word2. All the occurrences of Word1 in the text file are replaced by Word2.
5. **Quit:** The program ends.

The program is menu-driven (see figure below).

```
-----  
Enter your choice:  
1. Display specific Word Count  
2. Display all Unique Words  
3. Display all Word Counts  
4. Replace word  
5. Quit  
█
```

If the user chooses option 1, then a sample output is shown below.

```
-----  
Enter your choice:  
1. Display specific Word Count  
2. Display all Unique Words  
3. Display all Word Counts  
4. Replace word  
5. Quit  
1  
Enter word: the  
Word Count: 11
```

For option 2, print all the unique words present in the text separated by a semicolon (;)

```

-----
Enter your choice:
1. Display specific Word Count
2. Display all Unique Words
3. Display all Word Counts
4. Replace word
5. Quit
2
Unique Words:
; face. ; before ; multiply ; first ; The ; has ; cold ; back ; these ; is ; likely ; Over ; It'
s ; body ; now ; that ; contaminated ; (within ; COVID-19. ; because ; membrane ; from ; possible ; m
outh, ; nasal ; surface ; territories ; other ; there, ; tissue. ; 4,850,000 ; East ; transfer ; (10/
11/2021), ; your ; you ; or ; throat. ; common ; tissues. ; Middle ; released ; coughs, ; cause ; vir
uses ; COVID-19 ; a ; on ; They ; get ; continents ; illness ; contact ; shaking ; the ; spikes ; As
; hands ; hands. ; inhale ; hands) ; then ; may ; crown-like ; number ; sings ; than ; 1. ; through ;
examples ; Wuhan, ; spread: ; was ; touching ; 3. ; talks, ; 6 ; into ; But ; syndrome ; (touching,
; changes ; lung ; and ; enters ; It ; washing ; this ; daily. ; air ; Severe ; coronaviruses ; 2019.
; are ; can ; died. ; be ; attaches ; eyes, ; From ; all ; when ; infected ; droplets ; 196,910,000
; called ; been ; sneezes, ; - ; airborne ; in ; infected. ; face). ; breathes ; virus ; feet). ; per
son ; near ; to ; reported ; considered ; world ; 2. ; family ; China ; virus. ; moves ; unlikely. ;
people ; (directly ; cells ; begins ; acute ; Some ; Coronaviruses ; countries ; passages ; "corona"
; new ; nose ; (SARS), ; (MERS) ; cases ; spread ; continents. ; more ; after ; with ; respiratory ;
192 ; coronavirus ; close ; thought ; You ; since ; of ; strain ; an ; travels ; also ; writing ; Dec
ember ; it's ; humans. ; if ; eyes ; have ; mucous ; droplets.

```

For choice 3, the following is the output (cropped version):

```

-----
Enter your choice:
1. Display specific Word Count
2. Display all Unique Words
3. Display all Word Counts
4. Replace word
5. Quit
3
Word Counts:
Two : 1
members : 1
of : 4
the : 11
1984 : 1
class : 2
Jefferson : 1
High : 1

```

For choice 4, the user is required to enter 2 words, Word1 and Word2. All occurrences of Word1 are replaced by Word2 in the file.

```

-----
Enter your choice:
1. Display specific Word Count
2. Display all Unique Words
3. Display all Word Counts
4. Replace word
5. Quit
4
Enter word to be replaced: the
Enter word that will replace the: program
Replaced successfully!

```

- The input text file will be provided.
- Consider the following example regarding Option 2 and 3:
Let's say that the text file has the following contents: "the the is a break the if if"
Then the unique words in the text file are ["the", "is", "a", "break", "if"] (*and not ["is", "a", "break"]*).
So the output for Option 2 will be: the ; is ; a ; break ; if
For Option 3 the output becomes:
the : 3
is : 1
a: 1
break : 1
if : 2
- For Option 4, you have to rewrite the contents in the input file itself by replacing Word1 with Word2. (*No new file should be created*). The options can be chosen in any order and hence the results should be in accordance with the *updates* made by the user.
- In the final submission make sure that the input file is unaltered.

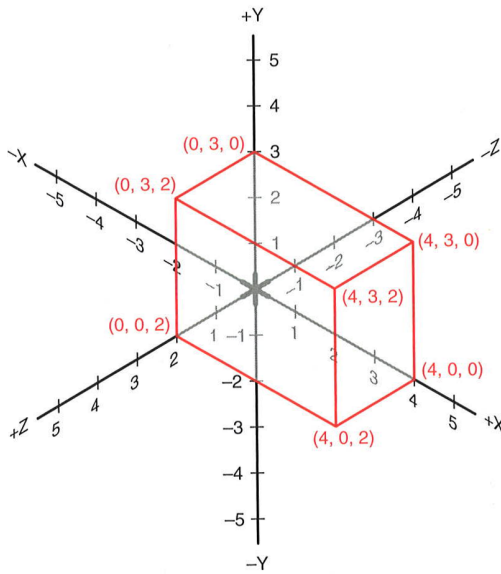
Q2. Transformations!!

In 3D graphics software like Blender, Maya, etc., we are able to create objects and perform some transformation operations on them. Making software like so is beyond the scope of this course but we can make a simple one that works on the console. Your task is to write a program that is able to create a 3D shape from given vertices and is able to perform 3 transformation operations which are Scaling, Translation, and Rotation.

Creating the 3D shape

The program takes as input three lists x, y, and z each of length n where (x[i], y[i], z[i]) represent the i^{th} vertex of the 3D Shape.

Example:



For the above cuboid

$$x = [0, 0, 0, 0, 4, 4, 4, 4]$$

$$y = [0, 0, 3, 3, 3, 0, 0, 3]$$

$$z = [0, 2, 2, 0, 0, 0, 2, 2]$$

Transformation

Any transformation T represented by a matrix (list of lists) can be applied on a point (x, y, z) to get the transformed point (x', y', z') .

A generic T can be represented as -

$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, applying T on point (x, y, z) is done using -

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

There are 3 types of transformations as mentioned earlier and each type of transformation has its own T matrix representation.

1. Scaling

For scaling, to scale the point by s_x , s_y and s_z ; T here is -

$$T = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Translating

For translating a point by t_x , t_y , and t_z ; T here, is -

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotation

The rotation can happen along 3 axes viz., x, y and z so each of these have different T matrix represented by $\text{rotate}_{\text{axes}}(\phi)$ where ϕ is the angle by which you wish to rotate the point.

$$\underbrace{\begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotate}_z(\phi)}, \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotate}_x(\phi)}, \underbrace{\begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotate}_y(\phi)}$$

Input:

The first line will contain **n** which is the number of vertices the 3D shape has

Next line will input the space-separated list **x**

Next line will input the space-separated list **y**

Next line will input the space-separated list **z**

Next line will input the number of transformation queries to apply which is **q**

Next **q** lines will input a query

Each query looks like -

For scaling, the query looks like

S sx sy sz

where, **S** represents scaling and **sx**, **sy** and **sz** represent the amount of scale in each axes

For translating, the query looks like

T tx ty tz

where, **T** represents translation and **tx**, **ty** and **tz** represent the amount of translation in each axes

For rotating, the query looks like

R x ϕ

where, **R** represents rotation and **x** represents the axis of rotation and **ϕ** represents the angle by which to rotate. Similarly, for y-axis we write **R y ϕ** and for z axis we write **R z ϕ**

Output:

Print the final transformed 3 lists x, y and z

Also, store these lists in a text file in the same format as input and output.

Q3. Conversion of numbers !!.

As we all know, we can do numbers representation in the different radix. On Planet Earth, humans use numbers with radix 10, and the computer uses numbers with radix two, etc. Scientists at NASA recently found that creatures living there may use different radix to represent a number on several other planets. For example, Creatures living on the Huru-Huru planet uses hexadecimal radix. On the other hand, creatures of the Zora planet uses octal representation. Scientists want to communicate with these creatures from a different world, so we need to convert numbers from one radix to another. For this task, NASA hired you. As a programmer at NASA, you are asked to create a menu-driven program for the following operations. For each process, **you must make appropriate functions**. For each of the following operations, you must ask a number represented in string as an

input and give the result in the other radix. You must handle it wisely if the user does not enter a number with the specified radix.

- 1) Convert decimal to binary and vice-versa
- 2) Convert decimal to hexadecimal and vice-versa
- 3) Convert decimal to octal and vice-versa.
- 4) Convert binary to hexadecimal and vice-versa.
- 5) Convert binary to octal and vice-versa.
- 6) Convert hexadecimal to octal and vice-versa.
- 7) Convert number with radix A to radix B. Here $A, B \leq 36$.

Example of Type 7) :

$$(GF12)_{18} = (98192)_{10}$$

$$(CA91)_{13} = (5C83)_{17}$$

Q4. Final Report!!

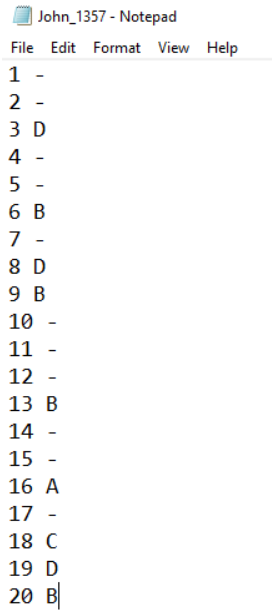
The National Testing Agency will not be able to hold this year's JEE paper. So, they have asked you to conduct the JEE paper but before that, they want to test you if you can write a program that can simulate a JEE-type environment.

The question paper has a total of 20 MCQ-type questions.

There are two folders representing the two high-level parties of the system which are "Student" and "Admin".

In the "Student" folder there will be text files with the naming convention "StudentName_StudentNo.txt". Each student has their own text file. Each text file represents the student's submission to the questions.

An example text file for the Student Name - John with Student Number - 1357 is

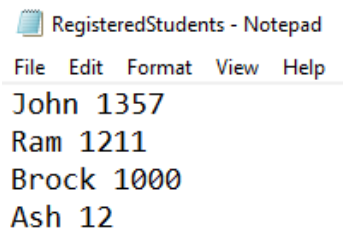


```
John_1357 - Notepad
File Edit Format View Help
1 -
2 -
3 D
4 -
5 -
6 B
7 -
8 D
9 B
10 -
11 -
12 -
13 B
14 -
15 -
16 A
17 -
18 C
19 D
20 B|
```

Where, “-” represents that the student didn’t attempt that question

There will be a “RegisteredStudents.txt” in the “Admin” folder.

Example of a RegisteredStudents.txt -



```
RegisteredStudents - Notepad
File Edit Format View Help
John 1357
Ram 1211
Brock 1000
Ash 12
```

Where, each line represents <StudentName> <StudentNo>

In the “Admin” folder there is also an “AnswerKey.txt” file which is the answer key to the question paper. Using this answer key, you need to evaluate the submissions of each student. And, finally report a “FinalReport.txt” which has the name, number, and score of each student.

Example of AnswerKey.txt -

AnswerKey - Notepad

File Edit Format View Help

1 B
2 B
3 D
4 A
5 B
6 B
7 A
8 C
9 C
10 B
11 A
12 A
13 C
14 A
15 B
16 D
17 B
18 B
19 C
20 D

Example of FinalReport.txt -

FinalReport - Notepad

File Edit Format View

John 1357 1
Ram 1211 60
Brock 1000 52
Ash 12 -2

Where, each line represents <StudentName> <StudentNo> <StudentScore>

The scores are evaluated as follows -

- If the student's answer matches the answer in the answer key they are awarded +4 marks
- If the student answered the question and it does not match the answer in the answer key they are awarded -1 marks
- If the student didn't answer the question they are awarded 0 marks.

Your task is to write a code that evaluates each student's submission and finally create a FinalReport.txt