Assignment 1

ME 781

(Basic Python Programming)

For the majority of this course, you will be working in Python notebooks on Google colab. Your LDAP ID should provide you access to G-suite and thus colab. Please familiarize yourself with the working of colab by logging in colab and going through the introductory resources. You will find a lot of resources on learning python and in general machine learning at colab.

We will expect you to upload colab notebooks when you are required to turn your programming assignments. Please make sure that you provide enough comments in the code for the TAs to understand that you know what you are coding.

We have a software to check plagiarism in code, if anyone found copying the code, strict action will be taken against such students and can also be given FR grade directly.

Assignment 1 instructions:

- Naming conventioin: Please name your colab files as **rollnumber_A1.ipyb**.
- Learn how to use markdown to add texts in colab to make your code interactive and understandable.
- After completion of your assignment download the file as .ipyn and submit it using Teams assignment tab.

1**.** Create a python code that will take an integer as an input and output all prime numbers less than the input number. The python code should be written from scratch without using any inbuilt libraries. However, you could later use SymPy libraries to compare your results.

Please make sure that the program is so written that it is able to handle any invalid input (example a negative integer, a float, a string, etc.).

Input Format

Define a subroutine/function that takes single input, where input could be any data type (string, integer, floating)

Hint: use --> **def** get_primenumbers(n)

Output Format

Generate 1 list of prime in ascending order

Sample Input 1

`9 (n = 9 integer)`

Sample Output 1

[2, 3, 5, 7]

Sample Input 2

`‘9’ (9 is a string)`

Sample Output 2

[2, 3, 5, 7]

Sample Input 3

`‘xyz’ (xyz is a string)`

Sample Output 3

‘You have entered wrong input’

Sample Input 4

`7.2 (7.2 is a float)`

Sample Output 4

[2, 3, 5, 7]

Sample Input 5

`3 (3 is an int)`

Sample Output 5

[2]

2. Solve the problem of the Tower of Hanoi using recursion for a variable number of rings which is provided by the user.

The famous Towers of Hanoi puzzle consists of 3 pegs (A, B, C) on one of which (A) are stacked n rings of different sizes, each ring resting on a larger ring. The objective is to move the n rings one by one until they are all stacked on another peg (B) in such a way that no ring is ever placed on a smaller ring; the other peg (C) can be used as a workspace.

The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where *n* is the number of disks.

Link: https://en.wikipedia.org/wiki/Tower_of_Hanoi

Input Format

Program should take an integer as input.

Let that integer be 'n', n = number of rings

Use flowing function:

n = int (input ('input = '))

Input Constraints:

n>0

n is a natural number

You should solve the problem in $2^n - 1$

Output Format

Generate 3 lists corresponding to each tower and modify list at each step

i.e. if n = 3, initial lists will be as follows:

A= [3,2,1] B= [] C= [] where, 3 means ring with higher diameter and so on,

[3, 2, 1] means 3 at bottom, 2 on top of 3 and 1 on top of 2

Sample Input: 3

Sample Output

```
Initial condition = [3,2,1] [] []
Step 1 = [3,2] [1] []
Step 2 = [3] [1] [2]
Step 3 = [3] [] [2,1]
```

Step 4 = [] [3] [2,1]

Step 5 = [1] [3] [2]

Step 6 = [1] [3,2] []

Step 7 = [] [3,2,1] []

Explanation

Three disk Hanoi problem link: https://www.youtube.com/watch?v=5QuiCcZKyYU