Assignment 4

Due Apr 5th Thu, 5:00PM

Instructions:

* This is a computer-based assignment (Total points: 8). Please write your code in Python.

(1) Submission: Your codes/programs must be submitted electronically by email to sbai@fau.edu by 5:00 PM on the above due date. Use your **fau.edu email address** to send the code (unless you do not have one). Your email header/subject line should be:

MAD2502-HW4-Name

Include the Python code as an attachment with filename (put all your functions/codes in this single Python file):

MAD2502-HW4-Name.py

(2) Naming: A comment at the top of your program file should also identify yourself and the assignment that you are submitting:

```
# Assignment 4
# Name:
# Z-Number:
```

Please also follow the requirement(s) after each question to *name* your functions.

(3) Python commands/libraries: Do not use Python libraries unless it is mentioned in the question. Do not use any Python pre-defined commands/libraries that have not been discussed in this class. For example, do not use the Python built-in command that computes the sum or computes the square root.

(4) Comments/citation: Please comment your code. All code should be written by you. If any part of what you turn in is not your own work – you learn it from books, webpages etc – the source must be referenced.

Question 1. (2 points) Simulate the following variant of birthday problem: First, a random day (say, 123) is fixed; then one starts to generate random days (e.g. within the range 1 to 365) until the same specific day (e.g. 123) occurs again. Denote n to be the number of random days generated so far.

Repeat the game for 5000 times and compute the average n over these games.

Question 2. (3 points) Simulate the 3-Birthday problem. Randomly generate birthdays (e.g. within the range 1 to 365) until the **same** birthday has occurred for 3 times. Denote n to be the number of random days generated so far.

Repeat the game for 10000 times and compute the average n over these games.

Question 3. (3 points) Use Sieve of Eratosthenes to output the **number** of primes less than or equal to x for $x = 2^4, 2^5, 2^6, \cdots, 2^{22}$ respectively.

Requirement: one should **only** call the Sieve of Eratosthenes function for **once**. Use a **loop** to consider all $x = 2^4, 2^5, 2^6, \cdots, 2^{22}$.