Assignment 2

Due Feb 26th Monday, 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-HW2-Name

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

MAD2502-HW2-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 2
# 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.

(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.

(5) Two random numbers

The following questions use your FAU Z-number (should be 8-digits). Please follow the instructions below to generate two random integers t and s. Denote your Z-number by z.

- If your Z-number starts with 0, truncate the leading consecutive zeros, e.g. 00020202 becomes 20202. Then $z = 20202$.
- If your Z-number does not start with 0, keep all of it. E.g. $z = 22222222$.

In either case, you have a number z now. First, load the *random* module in your code (e.g. put the following line on top of your program),

```
import random
```

Initialize the random seed using your z and then generate two random numbers t and s.

```
### Replace 22222222 by your number z as described above.
random.seed(22222222)
# Use 1e13 and 1e14 in the following line. Do not change them.
t = random.randint(1e13, 1e14)
# Use 1e7 and 1e8 in the following line. Do not change them.
s = random.randint(1e7, 1e8)
```

In this example, the random numbers $t = 45680048908090$ and $s = 63209945$ are generated. They will be used in the following questions. You will have different t and s generated depending on your Z-number.

Question 1. (2 points)
A perfect square is an integer that can be expressed as the product of two equal integers. For example, $4 = 2^2$ is a perfect square. An integer n is called **primitive** if it is divisible by no perfect square other than 1. For example, 15 is primitive but 12 is not, as 12 is divisible by $4 = 2^2$ (a perfect square).

Let t be the first random integer generated from your Z-number (see Page 2). Write a program to find the **largest even primitive** integer $n < t$.

Requirement: write a function with name "Largest_even_primitive",

```
### Solution to Question 1
def Largest_even_primitive(t):
        ... Your codes ...
```

The function should return the **largest even primitive** integer **strictly** smaller than t.

Hints: to detect if some n is a primitive number, one can try to divide n by **ALL** possible perfect squares x^2: if any of the remainders is 0, it implies n is not a primitive. As our n is large, it will be slow (e.g. try this yourself) to divide all x^2 from $x = 1$ to n. An important observation is that it is sufficient to divide all such x^2 upto $x = \lceil \sqrt{n} \rceil$ where $\lceil . \rceil$ is the ceiling function[1]. Thus it boils down to estimate the square root of n (for which we discussed on the class).

[1] https://en.wikipedia.org/wiki/Floor_and_ceiling_functions

Question 2. (2 points)
This question shows how to use the area of rectangles to estimate the definite integral of a function. Consider the definite integral,

$$\int_a^b f(x)\,\mathrm{d}x.$$

Divide the interval from $x = a$ to $x = b$ into n equal subintervals of length $\Delta x = \frac{b-a}{n}$ each. Let c_k be the midpoint in the k-th subinterval. For example, $c_1 = a + \frac{\Delta x}{2}$, $c_2 = a + \frac{3\Delta x}{2}$, $\cdots$ and $c_n = b - \frac{\Delta x}{2}$. Then the definite integral can be approximated by

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} f(c_i)\Delta x = \Delta x \cdot [f(c_1) + f(c_2) + \cdots + f(c_n)] .$$

Recall that this is just the midpoint Riemann sum in Calculus 1.

Now take $f(x) = x^2 + \frac{1}{2}$ and end points $a = 0$, $b = 10$. Approximate

$$\int_0^{10} \left(x^2 + \frac{1}{2}\right) \mathrm{d}x.$$

using the above method.

Requirement: define a function with name "Approx_integral",

```
### Solution to Question 2
def Approx_integral(n):
        ... Your codes ...
```

The input n to the function is the number of subintervals used to approximated the definite integral. The function should return the above sum $\sum_{i=1}^{n} f(c_i)\Delta x$. You should **choose the number** n such that,

$$\left| \sum_{i=1}^{n} f(c_i)\Delta x - \int_0^{10} \left(x^2 + \frac{1}{2}\right) \mathrm{d}x \right| \leq 0.001.$$

Write the number n explicitly in the code.

Note this implies that the error between the true value and the approximated value is tiny. The true value can be computed easily.

Question 3. (2 points)
Let s be the second random integer generated from your Z-number (see Page 2). Write a Python program that finds the approximated roots for the equation

$$(x-s)^2 = e^{(x-s)\cdot\cos(x-s)}$$

in the interval $x \in [s, s+15]$. Find **4 such roots** in the interval. Each root r should satisfy the error bound

$$\left|(r-s)^2 - e^{(r-s)\cdot\cos(r-s)}\right| \leq 0.01.$$

To use the cosine function in Python, load the *math* module in your code, e.g.,

```
from math import *
```

Then the cosine function can be called by "cos(x)".

Requirement: define a function with name "Roots_Q3",

```
### Solution to Question 3
def Roots_Q3():
        ... Your codes ...
```

You may use any method discussed in class. The requirement is to find at least **4 roots**.

Hints: search over the interval to identify all possible sub-intervals that contain the roots.

Question 4. (2 points)
Let s be the second random integer generated from your Z-number (see Page 2). Write a Python program that finds one approximated root for the function,

$$f(x) = x - \frac{x^3}{3} + \frac{2x^5}{15} - s.$$

Find any root r that satisfies the error bound

$$|f(r)| \leq 0.001.$$

Requirement: define a function with name "Roots_Q4",

```
### Solution to Question 4
def Roots_Q4():
        ... Your codes ...
```

Hints: You may use any method discussed in class. If some method does not work, try another one.