Updated 10/01/2019: All the negative differences were updated. The absolute value of the differences should be diplayed

CSE 231 **Fall 2019**

Computer Project #4

This assignment focuses on the design, implementation and testing of a Python program to calculate and display the values of selected mathematical functions (see below).

It is worth 40 points (4% of course grade) and must be completed no later than **11:59 PM on Monday, October 7 (two weeks because of the first exam).**

Assignment Overview

In this assignment, you will practice with functions.

Assignment Background

Many common mathematical functions can be expressed as the sum of a series of terms. In some cases (such as the sum of the first N natural numbers), the function can be expressed as a finite series of terms.

In other cases, the expression contains an infinite number of terms. For example, the tangent function can be expressed as the infinite series:

$$\tan x = \sum_{n=1}^{\infty} \frac{B_{2n}(-4)^n(1-4^n)}{(2n)!} x^{2n-1} = x + \frac{x^3}{3} + \frac{2x^5}{15} + \cdots \quad \text{for } |x| < \frac{\pi}{2}$$

(GIF courtesy of Wikipedia)

Assignment Specifications

You will develop a Python program which allows the user to select from a menu of options and which performs the requested calculations. You must use at least the four functions specified below. You are encouraged to use more functions of your own design. Global variables are not allowed. That is, any variable names used within any function must be parameters or be created in the function. You are not allowed to use advanced data structures such as list, dictionaries, classes, etc. However, you are allowed to read ahead and use try-except.

`sum_natural (N) ---→ int:`

a. This function accepts as the user input as numerical value and returns the sum of the first `N` natural numbers. If the user input is not a natural number (an integer > 0), the function should return `None`. Note the string method `.digit()` is helpful.

b. Parameters: `N (string)`

c. Returns : `int`

`approximate_euler () ---→ float:`

a. This function accepts no input and returns the approximate of the Euler number (The irrational number e that is the base of the natural logarithm). The number should be rounded to 10 decimal digits after the decimal point. The number is calculated as:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \cdots$$

The program will expand the series until the absolute value of the next term in the series is less than 1.0e-7 (the specified epsilon); that term will not be included in the sum.

b. Parameters: `no parameters`

c. Returns : `float`

`approximate_ sinh(x)---→ float:`

a. This function accepts as input a numeric value X (measured in radians) and returns the approximate value of the hyperbolic sine of X. The number should be rounded to 10 decimal digits after the decimal point. The number is calculated as:

$$\sinh x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots \quad \text{for all } x$$

The program will expand the series until the absolute value of the next term in the series is less than 1.0e-7 (the specified epsilon); that term will not be included in the sum.
The parameter is a string; if it isn't in the correct format for a float, return None.

b. Parameters: `x (str)`

c. Returns : `float` or `None`

`approximate_ cosh(x)---→ float:`

a. This function accepts as input a numeric value X (measured in radians) and returns the approximate value of the hyperbolic cosine of X. The number should be rounded to 10 decimal digits after the decimal point. The number is calculated as:

$$\cosh x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots \quad \text{for all } x$$

The program will expand the series until the absolute value of the next term in the series is less than 1.0e-7 (the specified epsilon); that term will not be included in the sum.
The parameter is a string; if it isn't in the correct format for a float, return None.

b. Parameters: `x (str)`

c. Returns : `float` or `None`

`main():`

a. This function is used to interact with the user. It takes no input and returns nothing. Call the functions from here. The program should prompt the user to choose between 6 options (capitalization does not matter) until the user enters `'X'`:

`'A'` - Display the value of the sum of the first N natural numbers.
`'B'` - Display the approximate value of e.
`'C'` - Display the approximate value of the hyperbolic sine of X.
`'D'` - Display the approximate value of the hyperbolic cosine of X.

`'M'` - Display the menu of options.
`'X'` - Exit from the program.

b. The program will repeatedly prompt the user to enter a letter (upper or lower case) which corresponds to one of the supported options. For example, the user will enter the letter A (or the letter a) to select Option A (display the value of the sum of the first N natural numbers).
c. The program will display the menu of options once when execution begins, whenever the user selects Option `M`, and whenever the user selects an invalid menu option.
d. If the user enters option `A`, the program will ask the user to enter a numeric value N. If N is a natural number, the program will calculate and display the value of the sum of the first N natural numbers. Otherwise, the program will display an appropriate message.
e. If the user enters option `B`, the program will calculate and display the approximate value of e. It will then display the corresponding value from the Python math module, as well as the absolute value of the difference between the calculated value and the math module value. The program should display 10 decimal digits after the decimal point for the approximate value. Use the appropriate string formatting.
f. If the user enters option `C`, the program should ask the user to enter a numeric value X (measured in radians). The program will calculate and display the approximate value of the hyperbolic sine of X. It will then display the corresponding value from the Python math module, as well as the absolute value of the difference between the calculated value and the math module value. The program should display 10 decimal digits after the decimal point for the approximate value. Use the appropriate string formatting.
g. If the user enters option `D`, the program will ask the user to enter a numeric value X (measured in radians). The program will calculate and display the approximate value of the hyperbolic cosine of X. It will then display the corresponding value from the Python math module, as well as the absolute value of the difference between the calculated value and the math module value. The program should display 10 decimal digits after the decimal point for the approximate value. Use the appropriate string formatting.
h. If the user quits, the program should display a goodbye message.

Assignment Notes and Hints

1. The coding standard for CSE 231 is posted on the course website:

   http://www.cse.msu.edu/~cse231/General/coding.standard.html

Items 1-9 of the Coding Standard will be enforced for this project.

2. The program will produce reasonable and readable output, with appropriate labels for all values displayed.

3. You may assume that the user enters a string representing a numeric value when prompted for a number. However, you should not assume that the value is valid in that particular context. For example, the user might enter 45.7 for Option A; that numeric value is not a natural number.

4. Be sure to prompt the user for the inputs in the correct order. And, your program cannot prompt the user for any supplementary inputs.

5. Several items from the math module might be useful for this project:

```
math.e
math.factorial()
math.sinh()
math.cosh()
```

6. The program will calculate the value for Option A using either a loop or an equation.

7. We provide a `proj04.py` program for you to start with.

8. You are not allowed to use advanced data structures such as list, dictionaries, classes, etc. However, you are allowed to read ahead and use try-except.

9. If you "hard code" answers, you will receive a grade of zero for the whole project. An example of hard coding is to simply print the approximate value of e rather than calculating it and then printing the calculated average.

10. There are multiple ways to check whether a string contains a float. One is to use the `float_check` function you developed in Lab 4. That function doesn't consider a float in the format such as `4e2`, but that is fine for this project—we will assume that the user will not use that format for input. However, the best way to check for a float is to use the `try-except` from Section 6.6.2 of the text. You `try` to convert a value to a `float` and if a `ValueError` exception is raised you handle it, e.g. `return None`. Feel free to read ahead and use the `try-except` statement in this project.

11. Good style defines a constant `EPSILON = 1.0e-7`

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Cycle through the following steps to incrementally develop your program:

    - Edit your program to add new capabilities.
    - Run the program and fix any errors.
    - Use the **Mimir** system to submit the current version of your program.

- Be sure to use the **Mimir** system to submit the final version of your program.

- Be sure to log out when you leave the room, if you're working in a public lab.

The last version of your solution is the program which will be graded by your TA.

You should use the ***Mimir*** *system to back up your partial solutions, especially if you are working close to the project deadline. That is the easiest way to ensure that you won't lose significant portions of your work if your machine fails or there are other last-minute problems.*

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj04.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir system** before the project deadline.

Test 1:

```
Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: n

Error:  unrecognized option [ N ]

Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: m

Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: x
Hope to see you again
```

Test 2:

```
Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: a


Enter N: 0

        Error: N was not a valid natural number [ 0 ]

Option: A


Enter N: 1.6

        Error: N was not a valid natural number [ 1.6 ]

Option: a


Enter N: 6

        The sum:  21

Option: x
Hope to see you again
```

Test 3:

```
Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: b

        Approximation: 2.7182818011
        Math module:   2.7182818285
        difference:     0.0000000274

Option: C


      Enter X: a

        Error: X was not a valid float [ a ]

Option: c


      Enter X: 1.5

        Approximation: 2.1292794235
        Math module:   2.1292794551
        difference:     0.0000000316

Option: C


      Enter X: 0

        Approximation: 0.0000000000
        Math module:   0.0000000000
        difference:    0.0000000000

Option: x
Hope to see you again
```

Test 4:

```
Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: d


     Enter X: kl

        Error: X was not a valid float [ kl ]

Option: 0

Error:  unrecognized option [ 0 ]

Please choose one of the options below:
        A. Display the value of the sum of the first N natural numbers.
        B. Display the approximate value of e.
        C. Display the approximate value of the hyperbolic sine of X.
        D. Display the approximate value of the hyperbolic cosine of X.
        M. Display the menu of options.
        X. Exit from the program.

Option: D


     Enter X: 0

        Approximation: 1.0000000000
        Math module:   1.0000000000
        difference:    0.0000000000

Option: d


     Enter X: 4

        Approximation:   27.3082328199
        Math module:     27.3082328360
        difference:       0.0000000161

Option: d


     Enter X: 1.5
```

```
        Approximation: 2.3524096119
        Math module:   2.3524096152
        difference:    0.0000000033

Option: x
Hope to see you again
```

Grading Rubric

```
Computer Project #04                             Scoring
Summary General Requirements:
  ( 4 pts) Coding Standard 1-9
     (descriptive comments, function headers, mnemonic identifiers,
     format, etc...)

Implementation:
 ( 4 pts)  sum_natural() function

 ( 5 pts)  approximate_euler() function

 ( 5 pts)  approximate_ sinh() function

 ( 5 pts)  approximate_ cosh() function

 ( 3 pts)  Test 1

 ( 4 pts)  Test 2

 ( 5 pts)  Test 3

 ( 5 pts)  Test 4



Note: hard coding an answer earns zero points for the whole project
-10 points for not using main()
```