

3. Homework

Due **9/24/13** at 11:55pm on Blackboard

Please create a separate python file for each of the two exercises below, and submit the files on Blackboard. Please use the following naming convention:

`lastName_firstName_hw3_number.py` and submit it on Blackboard; here, `number` should be either 1 or 2.

1. Sum of fractions (12 points)

- (a) (5 points) Write a function that takes `n` as input, and returns the sum

$$\sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots + \frac{1}{n} .$$

In the main body of your script, test your output with several values for `n`.
(*Hint: Make sure that you use floating point numbers to compute the sum. Your result should be greater than 1 for $n > 1$.*)

- (b) (3 points) Modify your function to include error handling of the following kind: If `n` is not an integer (for example a floating point number, or a string), return `-1`. In the main body of your script, test your output with several non-integer values for `n`.
- (c) (4 points) The following series is known to slowly approximate π :

$$\pi = 4 * \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} \dots \right)$$

Write a function that takes `n` as input and uses this formula to approximate π . Your function should compute a finite sum, and the number of summands should be specified using `n`. In the main body of your script, test your output with several *large* values for `n`

2. Conversion (10 points)

- (a) (5 points) Write a function that takes a list as input, and returns a list in reverse order. In the main body of your script, test your output with several values for `n`.
- (b) (5 points) Write a function that takes `n` as input, and converts `n` into a binary number. You can either print the output directly, represent it as a string, or represent it as a list of bits. In the main body of your script, test your output with several values for `n`. (*Hint: Use the conversion algorithm that repeatedly divides by 2 and uses the remainders to obtain the desired bits.*)