

Assignment #4

Practice Python and Compare Algorithms

Due: Sunday, 10/15/17, 11:59pm

(70 pts) We will now implement our simple binary calculator in Python using various strategies/algorithms **without any built-in functions**. This calculator will convert an unsigned decimal number less than 256 to a binary number, and you will compare and contrast the pros and cons of each strategy. For your assignment #3, you used one of these three strategies, which should make implementation a bit easier for that one, but you will need to provide a design for the steps/algorithm for the other two strategies in this assignment.

First Strategy: No conditionals/loops

Using this strategy, you are not allowed to use any conditional statements or loops.

- If you did not generate this algorithm in Assignment #3, then you must provide the design for the steps/algorithm used in this strategy.
- What are the pros and cons to this strategy?

Second Strategy: Only conditionals/no loops

Using this strategy, you are allowed to use conditional statements but you may not use any loops.

- If you did not generate this algorithm in Assignment #3, then you must provide the design for the steps/algorithm used in this strategy.
- What are the pros and cons to this strategy?
- How does it compare to the algorithm that didn't have conditionals?

Third Strategy: Conditionals and loops allowed

Using this strategy, you are allowed to use conditional statements and loops.

- If you did not generate this algorithm in Assignment #3, then you must provide the design for the steps/algorithm used in this strategy.
- What are the pros and cons to this strategy?
- How does it compare to the algorithm that didn't have conditionals and the one that did have conditionals?

(10 pts) Testing Table (Actual results):

- What does your program do when you enter good numbers?
- What does your program do when you enter a negative number, characters, a floating point number, or a number larger than 255?

(20 pts) Design

Design a solution to convert any positive number to an unsigned binary number. If your solution above using conditionals and loops works for any positive number, not just a number less than 256, then you must come up with an alternative solution/design to convert any positive number to binary.

(10 pts) Extra Credit: Implement Error Handling

Implement a solution for handling BAD input to your binary calculator without using try/catch exceptions. How are you going to make sure that your program prints an error message, "This is not a number 0-255!", when the user enters BAD input?

Electronically submit your **.py file** (Python code) and **designs/answers to questions as a .pdf** by the assignment due date, using TEACH:

https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth