* **[Assignment #1 - Error Correction](https://blackboard.newhaven.edu/webapps/assignment/uploadAssignment?content_id=_955876_1&course_id=_33511_1&group_id=&mode=view)**

Due: February 7, 2017

When data is transmitted between two computers, it is possible that the signal can encounter interference that would distort it, resulting in some of the data bits being changed. At the very least we would like to detect when such errors occur, and in some cases it is important that we be able to correct the errors. For instance, if we were transmitting data to a computer on a spaceship on Mars, we would not want to have to keep resending corrupted data due to the long travel delays. For this assignment you will write a program that is capable of performing a simple correction of an error in a transmitted data byte.

Assume we want to transmit 1 byte of data and be able to detect and correct any single bit that might be distorted during the transmission. In order to do this it is necessary to transmit extra bits of information along with the byte. The values of these bits are determined in such a way that particular calculations performed at the receiving end can determine if any errors occurred. The particular algorithm you will implement adds 4 extra bits. Assume that you start with a data byte where the 8 bits are labeled as:  d1 d2 d3 d4 d5 d6 d7 d8. The full twelve bit sequence that is transmitted will look like: c1 c2 d1 c4 d2 d3 d4 c8 d5 d6 d7 d8, where the 4 extra bits are denoted by c1, c2, c4, and c8.

When these 12 bits are received, a series of calculations needs to be done as follows.

1) Look at the data bits d1, d2, d4, d5 and d7 and count how many bits are 1's. If there is an odd number of 1's, the value of c1 should be a 1. If there is an even number of 1's, the value of c1 should be a 0. If the value of c1 is not what it should be, add the value 1 to a growing sum.

2) Perform the same check with data bits d1, d3, d4, d6, d7 and c2. If there is a mismatch, add the value 2 to the sum.

3) Perform the same check with data bits d2, d3, d4, d8 and c4. If there is a mismatch, add the value 4 to the sum.

4) Perform the same check with data bits d5, d6, d7, d8 and c8. If there is a mismatch, add the value 8 to the sum.

If the final sum value is 0, then no errors occurred.  If the sum value is greater than 12, then more than a single error occurred, which we cannot correct. For all other sum values, the number indicates which of the 12 bits was corrupted (they are numbered 1 - 12, going left to right), and therefore you change this value from a 0 to a 1, or a 1 to a 0 to correct the error.

Given the algorithm above, your program should do the following:

Input a string that should represent a 12 bit chunk of data that is received in a transmission. Check that the input is indeed 12 digits long. If not, print an error message. Also check whether each of the 12 digits is a 0 or 1. If there are any non-binary digits, print an error message. If the input was in error, ask the user to enter another value until they get it right.

Once valid input has been obtained, do the above calculations to determine what kind of errors might have occurred. If no error occurred, say so, and print out the correct 8 data bits in order. If a single error occurred, print out what data bit was detected as wrong, and then print out the correct 8 data bits in order. Finally, if too many errors occurred to correct, say so.

Put the entire processing of a bit sequence into a query loop, so that the user can process multiple data sequences.

You should properly comment your code, including putting your name in your source file. Any use of code from some source other than yourself should be cited in the comments. You should not use any aspects of the Python language beyond what we have discussed in class. You should submit your source code as well as sample output of your program that demonstrates its operation for both good and bad data.