# Problem 3: (Code review)

**Following are few recommendations for investigating the given program.**

1. Avoid using nested loop i.e., loop within a loop. The time complexity can easily become exponential with a nested loop.
2. Investigate the output – returning statements first:

The first two return statements involve constant values, which does not affect the time and space complexity of the program. However, the final return statement involves the concatenation of one constant value (2) and values of a range ***s*** (from 3 to n with 2 step increment).

In the final output, only ***s*** is a variable that has been utilized. However, in Python language a variable defined as range can only be initiated with a range it cannot be assigned with a new value. In this case, statements such as s[j] = 0 are erroneous.

Variable i, j, and m has no contribution in final output. Variable i is initiated with value 0 and then incremented every step. Finally, it does not contribute to the final returning statement or the output. Similarly, j is initialized every time within outer loop and later its value is incremented in the inner loop. However, j and m variables have no contribution in the final output.

Variable j has floating value due the division operation. A floating value can not be assigned as index of an element.

Conclusion:

The nested loops are independent with respect to the final output. So, they can be omitted to reduce time complexity of the given program.