Assertions and Exceptions-Exercise on Simple Divide

**Assertions and Exceptions-Exercise on Simple Divide:** Suppose we rewrite the FancyDivide function to use a helper function.

def fancy\_divide(list\_of\_numbers, index):

denom = list\_of\_numbers[index]

return [simple\_divide(item, denom) for item in list\_of\_numbers]

def simple\_divide(item, denom):

return item / denom

This code raises a ZeroDivisionError exception for the following call: fancy\_divide([0, 2, 4], 0)

Your task is to change the definition of simple\_divide so that the call does not raise an exception. When dividing by 0, fancy\_divide should return a list with all 0 elements. Any other error cases should still raise exceptions. You should only handle the ZeroDivisionError.

**Input Format:**

* The first line of the input contains the function fancy\_divide for which we pass an array(separated by comas) and an index as arguments

**Output Format:**

* Returns an array separated by comas.

**Sample Input #1:**

fancy\_divide([0, 2, 4], 0)

**Sample Output #1:**

[0, 0, 0]