

This problem is all about unit testing.

Your company needs a function that meets the following requirements:

- For a given array of n integers, the function returns the index of the element with the minimum value in the array. If there is more than one element with the minimum value, the returned index should be the smallest one.
- If an empty array is passed to the function, it should raise an Exception.

Note: The arrays are indexed from **0**.

A colleague has written that function, and your task is to design **3** separated unit tests, testing if the function behaves correctly. The implementation in Python is listed below (Implementations in other languages can be found in the code template):

```
def minimum_index(seq):
    if len(seq) == 0:
        raise ValueError("Cannot get the minimum value index from an empty sequence")
    min_idx = 0
    for i in range(1, len(seq)):
        if a[i] < a[min_idx]:
            min_idx = i
    return min_idx
```

Another co-worker has prepared functions that will perform the testing and validate returned results with expectations. Your task is to implement **3** classes that will produce test data and the expected results for the testing functions. More specifically: function `get_array()` in `TestDataEmptyArray` class and functions `get_array()` and `get_expected_result()` in classes `TestDataUniqueValues` and `TestDataExactlyTwoDifferentMinimums` following the below specifications:

- `get_array()` method in class `TestDataEmptyArray` has to return an empty array.
- `get_array()` method in class `TestDataUniqueValues` has to return an array of size at least 2 with all unique elements, while method `get_expected_result()` of this class has to return the expected minimum value index for this array.
- `get_array()` method in class `TestDataExactlyTwoDifferentMinimums` has to return an array where there are exactly two different minimum values, while method `get_expected_result()` of this class has to return the expected minimum value index for this array.

Take a look at the code template to see the exact implementation of functions that your colleagues already implemented.