Mock1

QuestionOne

Create a method `ManyTimes` that takes a string (`s`) and an int (`i`) as arguments and returns the string argument duplicated `i` many times. Your code should satisfy the following:

```
var m1 = ManyTimes("abc", 3);
Assert.AreEquals("abcabcabc", m1,"Your message here");
var m2 = ManyTimes("123", 2);
Assert.AreEquals("123123", m2, "Your message here");
```

replacing `"Your message here"` with an appropriate message.

Alert

Refactor the `AlertService` and `AlertDAO` classes from the file `Alert.cs` :

- Create a new interface, named IAlertDAO `, that contains the same methods as` AlertDAO`.
- `AlertDAO` should implement the `IAlertDAO` interface and overload appropriate methods from `Object` .
- `AlertService` should utilise constructor dependency injection for `IAlertDAO` .
- The `RaiseAlert` and `GetAlertTime` methods should use the same object that was injected.
- Provide appropriate tests to support your answer using `MSUnit` .

LastQuestion

- Write a program that traverses a collection of names (strings) and returns the first name whose sum of characters is equal to, or larger than, a given number `N` , which will be provided on the first input line.
- Use a function that accepts another function as one of its parameters. Start off by building a regular function to hold the basic logic of the program. Something along the lines of `Func<string, int, bool>` .
- Then create your main function which should accept the first function as one of its parameters to the program.

```
500
Fred Maxim Zad Matthew Mary
Result: Maxim
```