Test Script should be in a table format, with header as shown below. There should be **at least 3 distinct test classes (as indicated in the description) **per function**.

Test descriptions are supposed to be unique and should indicate classes/ groups of test cases on what is being tested. For example, given the function getAreaTri() which computes the area of a triangle given the base and height as parameters, the following are 3 distinct classes of tests:

- testing with base and height values smaller than 1
- testing with whole number values for base and height
- testing with floating-point number values for base and height, larger than 1

The following test descriptions are incorrectly formed:
Too specific: testing with base containing 0.25 and height containing 0.75
Too general: testing if function can generate correct area of triangle
Not necessary: since already defined in pre-condition: testing with base or height containing negative values

FILL-UP THE FOLLOWING TABLE (the 1st three rows are just examples; delete it in your own document). Add new rows as you deem necessary...

Function Name	**#**	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**Pass or Fail?**
RaiseTo	1	Base (x) and Exponent (n) are positive.	x = 2, n = 3	8	8.0000...	Pass
RaiseTo	2	Base (x) is positive and Exponent (n) is zero.	x = 2, n = 0	1	1.0000...	Pass
RaiseTo	3	Base (x) is positive and Exponent (n) is negative.	x = 2, n = -3	0.125	0.1250...	Pass
RaiseTo	4	Base (x) is zero and Exponent (n) is positive.	x = 0, n = 3	0	0.0000...	Pass
RaiseTo	5	Base (x) and Exponent (n) are zero.	x = 0, n = 0	1	1.0000...	Pass
RaiseTo	6	Base (x) is zero and Exponent (n) is negative.	x = 0, n = -3	undefined	nan	Pass
RaiseTo	7	Base (x) is negative and Exponent (n) is positive.	x = -2, n = 3	-8	-8.0000...	Pass
RaiseTo	8	Base (x) is negative and Exponent (n) is zero.	x = -2, n = 0	1	1.0000...	Pass
RaiseTo	9	Base (x) and Exponent (n) are negative.	x = -2, n = -3	-0.125	-0.1250...	Pass
factorial	1	Integer n is positive.	n = 5	120	120.0000...	Pass
factorial	2	Integer n is zero.	n = 0	1	1.0000...	Pass
factorial	3	Integer n is negative.	n = -5	undefined	nan	Pass
cosine	1	Theta (x) is positive.	x = PI / 2	0	0.0000...	Pass
cosine	2	Theta (x) is zero.	x = 0	1	1.0000...	Pass
cosine	3	Theta (x) is negative.	x = -(PI / 2)	0	0.0000...	Pass
sine	1	Theta (x) is positive.	x = PI / 2	1	1.0000...	Pass
sine	2	Theta (x) is zero.	x = 0	0	0.0000...	Pass
sine	3	Theta (x) is negative.	x = -(PI / 2)	-1	-1.0000...	Pass