Range Test

Initialize values => Lower Bound =1

=>Upper Bound = 10

1-Method (getLowerBoundary())

Test Case Number	In Put	Out Put (expected)
1	-	1.0
2	-	2.0
3	-	1.001
4	-	-1.0
5	-	0

Equitant Test Case		Boundary (range only is the lower bound value=1.0)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 4	Test case number 1	Test case number 1	Test case number 2	Test case number 4
			Test case number 3	Test case number 5

2-Method (getUpperBound())

Test Case Number	In Put	Out Put(expected)
1	-	10
2	-	12
3	-	10.0055
4	-	-10
5	-	9.75

Equitant Test Case		Boundary (range only is the Upper bound value=10.0)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 4	Test case number 1	Test case number 1	Test case number 2	Test case number 4
			Test case number 3	Test case number 5

3-getLength()

Test Case Number	In Put	Out Put (expected)
1	-	10
2	-	-10
3	-	10.025
4	-	9.75

Equitant Test Case		Boundary (range only is the length value=10)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 3	Test case number 2
				Test case number 4

4-**getCentralValue**()

Test Case Number	In Put	output Expected
1	-	5
2	-	-5
3	-	4.099
4	-	5.0025

Equitant Test Case		Boundary (range only is the center value=5)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 4	Test case number 1
				Test case number 3

5-**contains()**

Test Case Numbr	In put	Output expected
1	5	True
2	11	False
3	1	True
4	10	True
5	0	False
6	10.5	False
7	0.5	False

Equitant Test Case		Boundary (range only is the range value=[1,2,3,4,5,6,7,8,9,10])		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 2	Test case number 5
		Test case number 3	Test case number 6	Test case number 7
		Test case number 4		

Data Utilities Test

Initialization

1-Value2D (V)

5	9	14
10	15	14
9	12	25

1-Method(`calculateColumnTotal()`)

Test Case Number	In Put	Out Put (expected)
1	(data,0)	24
2	(data,0)	-24
3	(data,0)	26
4	(data,0)	20
5	(Null,0)	0

Equitant Test Case		Boundary (range only the sum of Colum (0) value=24)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 3	Test case number 2
Test case number 5				Test case number 4

2-`calculateRowTotal()`

Test Case Number	In Put	Out Put (expected)
1	(data,0)	28
2	(data,0)	-28
3	(data,0)	30
4	(data,0)	26
5	(null,0)	0

Equitant Test Case		Boundary (range only the sum of Colum (0) value=24)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 3	Test case number 2
Test case number 5				Test case number 4

3-*createNumberArray()*

```
Number arrnum[]=new Number[4];
Number arrnum2[]=new Number[4];
Number arrnum3[]=new Number[4];
Number arrnum4[]=new Number[4];
double arrdata[]=new double[4];
double arrdata2[]=null;
public void prepararr1D()
    {
    for (double i=0;i<4;i++)
        arrdata[(int) i]=i;
    for (double i=0;i<4;i++)
        arrnum[(int) i]=i;
    for (double i=0;i<4;i++)
        arrnum2[(int) i]=-i;
    for (double i=0;i<4;i++)
        arrnum3[(int) i]=i+1;
    for (double i=0;i<4;i++)
        arrnum4[(int) i]=i-1;
    }
```

Test Case Number	In Put	Out Put (expected)
1	(arrdata)	Arrnum
2	(arrdata)	Arrnum2
3	(arrdata)	Arrnum3
4	(arrdata)	Arrnum4
5	(arrdata2)	null

Equitant Test Case		Boundary (range only the number in arrdata)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 3	Test case number 2
Test case number 5				Test case number 4

4- *createNumberArray2D()*

```
Number arrnum2d[][]=new Number[4][2];
Number arrnum2d2[][]=new Number[4][2];
Number arrnum2d3[][]=new Number[4][2];
Number arrnum2d4[][]=new Number[4][2];
double arrdata2d[][]=new double[4][2];
double arrdata2d2[][]=null;
public void prepare2DArray()
    {
        for(double i =0;i<4;i++)
        {
            for (double j=0;j<2;j++)
                arrdata2d[(int) i][(int) j]=i+j;
        }
        for(double i =0;i<4;i++)
        {
            for (double j=0;j<2;j++)
                arrnum2d[(int) i][(int) j]=i+j;
        }
        for(double i =0;i<4;i++)
        {
            for (double j=0;j<2;j++)
                arrnum2d2[(int) i][(int) j]=-i+j;
        }
        for(double i =0;i<4;i++)
        {
            for (double j=0;j<2;j++)
                arrnum2d3[(int) i][(int) j]=i+j+1;
        }
        for(double i =0;i<4;i++)
        {
            for (double j=0;j<2;j++)
                arrnum2d4[(int) i][(int) j]=i+j-1;
        }
    }
```

Test Case Number	In Put	Out Put (expected)
1	(arrdata2d)	Arrnum2d
2	(arrdata2d)	Arrnum2d2
3	(arrdata2d)	Arrnum2d3
4	(arrdata2d)	Arrnum2d4
5	(arrdata2d2)	Null

Equitant Test Case		Boundary (range only number in arrdata2d)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 3	Test case number 2
Test case number 5				Test case number 4

5-*getCumulativePercentages()*

```
DefaultKeyedValues key=new  DefaultKeyedValues();
DefaultKeyedValues key2=new  DefaultKeyedValues();
DefaultKeyedValues key3=new  DefaultKeyedValues();
DefaultKeyedValues key4=null;
DefaultKeyedValues key5=new  DefaultKeyedValues();
public void preparekry()
{
    key.addValue("0",5);
    key.addValue("1",9);
    key.addValue("2",2);
}
public void preparekry2()
{
    key.addValue("0",0.3125);
    key.addValue("1",0.875);
    key.addValue("2",  1.0);
}
public void preparekry3()
{
    key.addValue("0",-0.3125);
    key.addValue("1",0.875);
    key.addValue("2", - 1.0);
}
public void preparekry5()
{
    key.addValue("0",3.3125);
    key.addValue("1",1.875);
    key.addValue("2",  2.0);
}
```

Test Case Number	In Put	Out Put (expected)
1	(key)	Key2
2	(key)	Key3
3	(key4)	0
4	(Key)	Key5

Equitant Test Case		Boundary (range only the values of return key)		
Invalid	Valid	In Range	More than Range	Less than Range
Test case number 2	Test case number 1	Test case number 1	Test case number 4	Test case number 2
Test case number 3				