NumberUtilNextClosesFibonacciTrueTest

```
package org.csystem.util.numeric;

import org.csystem.util.numeric.data.IntIntDataInfo;
import org.junit.Assert;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

import java.util.Collection;
import java.util.List;

@RunWith(Parameterized.class)
@Ignore("Tested before and passed")
public class NumberUtilNextClosesFibonacciTrueTest {

    public IntIntDataInfo intIntDataInfo;


    public NumberUtilNextClosesFibonacciTrueTest(IntIntDataInfo intIntDataInfo)
    {
        this.intIntDataInfo = intIntDataInfo;
    }


    @Parameterized.Parameters
    public static Collection<IntIntDataInfo> createData()
    {
        return List.of(new IntIntDataInfo(2,3),new IntIntDataInfo(5,8),new IntIntDataInfo(89,144));
    }

    @Test
    public void Test()
    {
        Assert.assertEquals(intIntDataInfo.expected,NumberUtil.nextClosestFibonacciNumber(intIntDataInfo.input));
    }

}
```

NumberUtilNextClosesFibonacciFalseTest

```
package org.csystem.util.numeric;


import org.csystem.util.numeric.data.IntIntDataInfo;
import org.junit.Assert;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

import java.util.Collection;
import java.util.List;

@RunWith(Parameterized.class)
public class NumberUtilNextClosesFibonacciFalseTest {

    public IntIntDataInfo intIntDataInfo;

    public NumberUtilNextClosesFibonacciFalseTest(IntIntDataInfo intIntDataInfo)
    {
        this.intIntDataInfo= intIntDataInfo;
```

```
    }


    @Parameterized.Parameters
    public static Collection<IntIntDataInfo> createData()
    {
        return List.of(new IntIntDataInfo(1,2),new IntIntDataInfo(3,4),new IntIntDataInfo(5,6));
    }


    @Test
    public void test()
    {
        Assert.assertNotEquals(intIntDataInfo.expected,intIntDataInfo.input);
    }
}
```

NumberUtilsFactorialTrueTest

```
package org.csystem.util.numeric;

 import org.csystem.util.numeric.data.IntLongDataInfo;
 import org.junit.Assert;
 import org.junit.Ignore;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

 import java.util.Collection;
 import java.util.List;

@Ignore("Tested before and passed")
@RunWith(Parameterized.class)
public class NumberUtilsFactorialTrueTest {

    public IntLongDataInfo intLongDataInfo;

    public NumberUtilsFactorialTrueTest(IntLongDataInfo intLongDataInfo)
    {
        this.intLongDataInfo = intLongDataInfo;
    }


    @Parameterized.Parameters
    public static Collection<IntLongDataInfo> creatData()
    {
        return List.of(new IntLongDataInfo(3,6L),new IntLongDataInfo(5,120L), new IntLongDataInfo(13, 6227020800L),new IntLongDataInfo(4, 2
    }


    @Test
    public void test()
    {
        Assert.assertEquals(intLongDataInfo.expected, NumberUtil.factorial(intLongDataInfo.input));
    }
}
```

NumberUtilsFactorialFalseTest

```
package org.csystem.util.numeric;
```

```
import org.csystem.util.numeric.data.IntLongDataInfo;
import org.junit.Assert;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

import java.util.Collection;
import java.util.List;

@RunWith(Parameterized.class)
public class NumberUtilsFactorialFalseTest {

    public IntLongDataInfo intLongDataInfo;

    public NumberUtilsFactorialFalseTest(IntLongDataInfo intLongDataInfo)
    {
        this.intLongDataInfo = intLongDataInfo;
    }

    @Parameterized.Parameters
    public static Collection<IntLongDataInfo> createData()
    {
       return List.of(new IntLongDataInfo(2, 4L),new IntLongDataInfo(3,6L),new IntLongDataInfo(5,120L));
        //Burada bilerek doğru değerler verilmiştir.

    }

    @Test
    public void test()
    {
        Assert.assertNotEquals("Cevap Hatalı", intLongDataInfo.expected, NumberUtil.factorial(intLongDataInfo.input));
    }

}
```