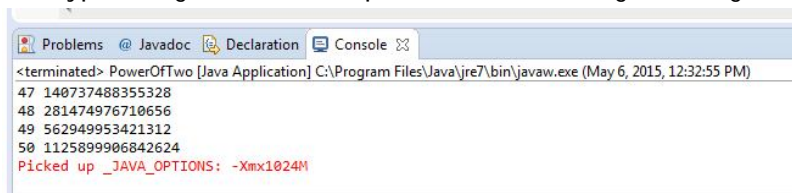


1. Use and adapt the code PowersOfTwo.java, to print the first 50 powers of 2^N . Include your code as well as the output result.

```
/******  
* Compilation: javac PowersOfTwo.java  
* Execution:   java PowersOfTwo N  
*  
* This program takes a command-line argument N and prints a table of  
* the powers of 2 that are less than or equal to  $2^N$ .  
*  
* Remarks  
* -----  
* Only works if  $0 \leq N < 63$  since  $2^{63}-1$  overflows an Long.  
*  
*****/
```

```
public class PowersOfTwo {  
    public static void main(String[] args) {  
  
        // read in one command-line argument  
        int N = Integer.parseInt(args[0]);  
  
        int i = 0;           // count from 0 to N  
        long powerOfTwo = 1; // change to long in order to handle the first 62 powers  
        // repeat until i equals N  
        while (i <= N) {  
            System.out.println(i + " " + powerOfTwo); // print out the power of two  
            powerOfTwo = 2 * powerOfTwo;              // double to get the next one  
            i = i + 1;  
        }  
    }  
}
```

Datatype changed on variable powerOfTwo from integer to long in order to handle the first 62 powers.



```
<terminated> PowerOfTwo [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 6, 2015, 12:32:55 PM)  
47 140737488355328  
48 281474976710656  
49 562949953421312  
50 1125899906842624  
Picked up _JAVA_OPTIONS: -Xmx1024M
```

3. Use the code Factors.java that prints the prime factors of a number. Follow the examples in the code headings comments and you are required to measure the computation time for the next 6 cases: 3, 6, 9, 12, 15, and 18 digit primes

Results in milliseconds

java Factors 997

```
The prime factorization of 997 is: 997
Elapsed Time: 0
Picked up _JAVA_OPTIONS: -Xmx1024M
```

java Factors 999983

```
The prime factorization of 999983 is: 999983
Elapsed Time: 0
Picked up _JAVA_OPTIONS: -Xmx1024M
```

java Factors 999999937

```
The prime factorization of 999999937 is: 999999937
Elapsed Time: 2
Picked up _JAVA_OPTIONS: -Xmx1024M
```

java Factors 999999999989

```
The prime factorization of 999999999989 is: 999999999989
Elapsed Time: 21
Picked up _JAVA_OPTIONS: -Xmx1024M
```

java Factors 999999999999989

```
The prime factorization of 999999999999989 is: 999999999999989
Elapsed Time: 359
Picked up _JAVA_OPTIONS: -Xmx1024M
```

java Factors 99999999999999989

```
44 public class Factors {
45
46     public static void main(String[] args) {
47         long startTime = System.currentTimeMillis();
48         // command-line argument
49         long n = Long.parseLong(args[0]);
50
51         System.out.print("The prime factorization of " + n + " is: ");
52
53         // for each potential factor i
54         for (long i = 2; i*i <= n; i++) {
55
56             // if i is a factor of N, repeatedly divide it out
57             while (n % i == 0) {
58                 System.out.print(i + " ");
59                 n = n / i;
60             }
61         }
62
63         // if biggest factor occurs only once, n > 1
64         if (n > 1) System.out.println(n);
65         else      System.out.println();
66
67         long stopTime = System.currentTimeMillis();
68         long elapsedTime = stopTime - startTime;
69         System.out.println(elapsedTime);
70     }
71 }
72
```

Problems @ Javadoc Declaration Console

<terminated> Factors [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 6, 2015, 11:21:17
The prime factorization of 99999999999999989 is: 99999999999999989
10726
Picked up _JAVA_OPTIONS: -Xmx1024M

4. Use the program FunctionGrowth.java that prints a table of the values of $\log N$, N , $N \log N$, N^2 , N^3 , and $2N$ for $N = 16, 32, 64, \dots, 2048$. What are the limits of this code? Suppose we want to stop not at $N=2048$. but at $N=1073741824$. Modify your code to do this. Add the modified code to your document and include generated output.

Limits, variable printed out is a long it will overflow, to handle a full table for 1073741824 you need to change variable to double.

```

22
23 public class FunctionGrowth {
24
25     public static void main(String[] args) {
26         System.out.println("log N \tN \tN log N\tN^2 \tN^3");
27         for (double i = 2; i <= 1073741824; i *= 2) {
28             System.out.print((int) Math.Log(i));
29             System.out.print('\t');           // tab character
30             System.out.print(i);
31             System.out.print('\t');
32             System.out.print((int) (i * Math.Log(i)));
33             System.out.print('\t');
34             System.out.print(i * i);
35             System.out.print('\t');
36             System.out.print(i * i * i);
37             System.out.println();
38         }
39     }
40 }
41

```

Problems @ Javadoc Declaration Console

<terminated> FunctionGrowth [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 6, 2015, 11:42:05 PM)

log N	N	N log N	N^2	N^3
0	2.0	1	4.0	8.0
1	4.0	5	16.0	64.0
2	8.0	16	64.0	512.0
2	16.0	44	256.0	4096.0
3	32.0	110	1024.0	32768.0
4	64.0	266	4096.0	262144.0
4	128.0	621	16384.0	2097152.0
5	256.0	1419	65536.0	1.6777216E7
6	512.0	3194	262144.0	1.34217728E8
6	1024.0	7097	1048576.0	1.073741824E9
7	2048.0	15615	4194304.0	8.589934592E9
8	4096.0	34069	1.6777216E7	6.8719476736E10
9	8192.0	73817	6.7108864E7	5.49755813888E11
9	16384.0	158991	2.68435456E8	4.398046511104E12
10	32768.0	340695	1.073741824E9	3.5184372088832E13
11	65536.0	726817	4.294967296E9	2.81474976710656E14
11	131072.0	1544487	1.7179869184E10	2.251799813685248E15
12	262144.0	3270678	6.8719476736E10	1.8014398509481984E16
13	524288.0	6904766	2.74877906944E11	1.44115188075855872E17
13	1048576.0	14536349	1.099511627776E12	1.15292150460684698E18
14	2097152.0	30526334	4.398046511104E12	9.223372036854776E18
15	4194304.0	63959939	1.7592186044416E13	7.378697629483821E19
15	8388608.0	133734419	7.0368744177664E13	5.9029581035870565E20
16	1.6777216E7	279097919	2.81474976710656E14	4.722366482869645E21
17	3.3554432E7	581453998	1.125899906842624E15	3.777893186295716E22
18	6.7108864E7	1209424316	4.503599627370496E15	3.022314549036573E23
18	1.34217728E8	2147483647	1.8014398509481984E16	2.4178516392292583E24
19	2.68435456E8	2147483647	7.2057594037927936E16	1.9342813113834067E25
20	5.36870912E8	2147483647	2.8823037615171174E17	1.5474250491067253E26
20	1.073741824E9	2147483647	1.15292150460684698E18	1.2379400392853803E27

Picked up _JAVA_OPTIONS: -Xmx1024M

I found that for handling big numbers, more than double, you can use special Java class: BigInteger, I didn't know how to implement on this example.

5. Modify the code Binary.java that converts any number to binary form, to convert any number to its hexadecimal form. Print the first 256 numbers in hex. Include code and output in your working document.

```
3 public class Binary {
4     public static void main(String[] args) {
5
6         // read in the command-line argument
7         char[] hexDigits = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
8                             'A', 'B', 'C', 'D', 'E', 'F'};
9         for (int i=1; i<=256; i++){
10             int n = i;
11             String hex = "";
12             while (n != 0) {
13                 int rem = n % 16;
14                 hex = hexDigits[rem] + hex;
15                 n = n / 16;
16             }
17             System.out.println(hex);
18         }
19     }
20 }
21
22 }
23 }
24 }
```

Problems @ Javadoc Declaration Console

<terminated> Binary [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 7, 2015, 12:00:27 AM)

E4
E5
E6
E7
E8
E9
EA
EB
EC
ED
EE
EF
F0
F1
F2
F3
F4
F5
F6
F7
F8
F9
FA
FB
FC
FD
FE
FF
100
Picked up _JAVA_OPTIONS: -Xmx1024M

6. Modify the code DayOfWeek.java to print the Day of the Week (Sunday, Monday, ...).

String array added with names of days, the printed out will be the d0 variable minus 1 as index of the String array.

```
24 public class DayOfWeek {
25     public static void main(String[] args) {
26         int m = Integer.parseInt(args[0]);
27         int d = Integer.parseInt(args[1]);
28         int y = Integer.parseInt(args[2]);
29
30         int y0 = y - (14 - m) / 12;
31         int x = y0 + y0/4 - y0/100 + y0/400;
32         int m0 = m + 12 * ((14 - m) / 12) - 2;
33         int d0 = (d + x + (31*m0)/12) % 7;
34         String[] daysName = {"Monday", "Tuesday", "Wed", "Thursday", "Friday", "Sat", "Sun"};
35
36         System.out.println(daysName[d0-1]);
37     }
38 }
```

Problems @ Javadoc Declaration Console

<terminated> DayOfWeek [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 7, 2015, 12:14:26 AM)

Thursday

Picked up _JAVA_OPTIONS: -Xmx1024M

The above result is the day for 5 7 2015