Computer Science II Exam #1

Date: 6/12/2014

Last Name: ___________________, First Name: ___________________

1) (20 pts) You are given a composite number n, but are asked to give proof that it is composite. You have a test that you can run such that, given a composite number, it proves that it is composite 50% of the time. Thus, to achieve your goal, you'll simply keep on repeating your test until you obtain the proof that the given number is composite. Fill out the chart below indicating the probability your algorithm will run the test various number of times.

Number of Times Test is Run	Probability
1	½
2	
3	
4	
k	

Using the chart above, write an infinite summation, in summation notation equaling the expect number of times your test will have to be run to prove that a given composite number n is composite.

Solve the summation on the following page. You should get a constant value as your answer. (Use the scratch page if you need more room to complete the summation.) Also, no need to use summation notation here. Feel free to write out the pattern of the sum so that it's easier to see your work.

2) (20 pts) Consider the following program description. Complete the code on the following page so it solves the problem.

A k-gap sequence of integers $a_1, a_2, \ldots, a_n$ is such that for every i, $1 \le i \le n-1$, $|a_i - a_{i+1}| \ge k$. For example, the sequence 2, 8, 3, 6, 1, 7 is a 3-gap sequence but not a 4-gap sequence, since the difference the consecutive terms, 3 and 6 is 3.

Write a program that reads in a sequence of integers and an integer k and outputs the permutation of the input sequence that is first lexicographically that is a valid k-gap sequence.

Input

The first line of the input file will contain two positive integers, n *(*$1 \le n \le 12$*)*, and k *(*$1 \le k \le 10^6$*)*, where n represents the length of the input sequence for which we are looking for the ordering that is a valid k-gap sequence that is first, lexicographically. The next line will contain the *n* integers.

Output

Output the sequence specified by the problem description with each integer separated by a comma and a space.

Sample Input

```
5 10
13 19 3 2 17
```

Sample Output

```
13, 3, 19, 2, 17
```

```
import java.util.*;

public class gap {

    public static int diff;
    public static int[] values;

    public static void main(String[] args) {
        Scanner stdin = new Scanner(System.in);
        int n = stdin.nextInt();
        diff = stdin.nextInt();
        values = new int[n];
        for (int i=0; i<n; i++)
            values[i] = stdin.nextInt();
        Arrays.sort(values);
        printSeq(new int[n], 0, new boolean[n]);
    }

    public static void print(int[] arr) {
        for (int i=0; i<arr.length-1; i++)
            System.out.print(arr[i]+", ");
        System.out.println(arr[arr.length-1]);
    }
```

```
        public static boolean printSeq(int[] cur, int k, boolean[] used) {

            if (k == cur.length) {
                print(cur);
                return true;
            }

            boolean ans = false;
            for (int i=0; i<used.length; i++) {


                /*** FILL IN ALL OF  YOUR CODE HERE ***/


            }

            return ans;
        }

    }
```

3) (9 pts) In the O(n) solution to the MCSS problem, there is a single running sum that is tallied as the algorithm iterates through each number in the input sequence. Below, write that running sum ***after*** the ith number has been processed.

Sequence	3	8	-5	2	-10	1	6	-8	12
Sum									

4) (10 pts) Show the state of the array storing a disjoint set AFTER it had undergone the following operations.(Note: Assume that the items in the disjoint set of size n are 0 through n-1.) Assume that the shorter tree is always attached to the longer tree and that if two trees of equal height are put together that the tree with the higher root value is attached to the tree with the lower root value.

```
DisjointSet dj = new DisjointSet(10);
dj.union(3, 8);
dj.union(2, 4);
dj.union(8, 7);
dj.union(4, 7);
dj.union(9, 1);
dj.union(9, 6);
dj.union(3, 0);
dj.union(5, 9);
```

Index	0	1	2	3	4	5	6	7	8	9
Value										

5) (4 pts) Show the final result of inserting the value 47 into the 2-4 tree depicted below.

```
          10,       20,         40
        /         |          |          \

3, 5      15, 17    31, 33       41, 50, 60
```

6) (8 pts) Show the result of running Radix Sort on the following 5 letter words (which are to be sorted in alphabetical order):

Original List	1st iteration	2nd iteration	3rd iteration	4th iteration	Sorted List
`mouse`					
`homer`					
`zebra`					
`house`					
`rainy`					
`tease`					
`green`					
`towel`					
`horse`					

7) (10 pts) Using induction on n, prove that for all n > 3, that it takes more than n comparisons to sort n numbers, using a comparison based sort.

8) (9 pts) On the Sudoku assignment, the following test case took too much time on a majority of the submissions:

```
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 9 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 9
```

Why would this case in particular take a long time to run, using a standard backtracking solution? What is a relatively simple “fix” to the code that would allow for this test case to be processed quickly?

9) (9 pts) Several faulty submissions to the Underground Cables assignment had an edge class similar to the following

```
class edge implements Comparable<edge> {

      public int v1;
      public int v2;
      public double distance;

      public edge(int start, int end, double d) {
            v1 = start;
            v2 = end;
            distance = d;
      }

      public int compareTo(edge other) {
            return (int)(this.distance – other.distance);
      }
}
```

Why is this code potentially faulty? Create a small test case of three vertices for which this code may cause an incorrect result from an implementation of Kruskal's that uses it.

10) (1 pt) Name one food item you can order at Del Taco. ____________________________

Scratch Page – Please clearly mark any work on this page you would like graded.