Input and Output format

The input for all the programs must be read from .txt files. The detailed format for each problem is given in the below.

1. Given two sets A and B, write a program that performs the following tasks:
   1. Performs intersection of two sets A and B
   2. Performs union of two sets A and B
   3. Finds the set difference A – B.

**Input Format:** The input file (.txt format) contains two rows, each row represents the information about a set (first row for A and second row for B).

The first element in row denotes the size of the set and the rest of the (distinct) elements are members of the set.

**Output Format:** You need to print three rows, one for each operation and each row is a collection of distinct elements in the increasing order.

1. Bob is a naughty kid and has recently shifted to his new house. He has a staircase in his house which has n-steps. He thinks of climbing the stairs with either 1 step or 2 steps at a time. Considering the initial position of Bob to be in the bottom, write a program to return the number of ways in which Bob can reach the top.

**Input Format:** The input file contains one positive integer denoting the number of steps in the staircase.

**Output Format:** Your output must be a single number denoting the number of possible ways Bob can climb to the top of the staircase.

1. Input: 2  
   Expected Output: 2
2. Input: 3  
   Expected Output: 3
3. Input: 0  
   Expected Output: 0
4. Input: 1  
   Expected Output: 1
5. Input: 10  
   Expected Output: 89
6. Input: 5  
   Expected Output: 8
7. Problem 3: Given a function (0-1 matrix) on set {1, 2, 3, …, n}, write a program to check it is one-to-one and onto function.

**Input Format:**

The file contains lines. The first line contains a positive integer which denotes the size of the matrix. The next rows and columns represent the 0-1 matrix. The elements in each row are separated by a space.

**Output Format:**

If the function is one-to-one, then output 1 else 0. Similarly, if the function is onto output 1 else print 0.

Your final output must be a two-digit binary number. The first digit is the answer for one-to-one and the second digit is the output for onto function.

Test-case:

Input:

3

1 0 0

0 1 0

0 0 1

Output:

11

1. In the class, we have discussed that the set of rational numbers is countable, by giving an ordering of elements of the set (the ordering contains every element exactly once). Write a program to find the position of given rational number where and are integers such that . Further, note that the terms and are need not be minimum terms i.e., , like 255/366.

**Input Format:** The input file contains two positive integers separated by a space. The first number denotes and the second number denotes .

**Output Format:** Your output must be a single number denoting the position of the rational number in the order.

First 20 numbers in the order as given below for your reference:

position= 1, number= 1/1

position: 2, number= 1/2

position: 3, number= 2/1

position= 4, number= 3/1

position= 5, number= 1/3

position: 6, number= 1/4

position: 7, number= 2/3

position: 8, number= 3/2

position: 9, number= 4/1

position= 10, number= 5/1

position= 11, number= 1/5

position: 12, number= 1/6

position: 13, number= 2/5

position: 14, number= 3/4

position: 15, number= 4/3

position: 16, number= 5/2

position: 17, number= 6/1

position= 18, number= 7/1

position= 19, number= 5/3

position= 20, number= 3/5

1. Given a partial order relation , find the topological order of the elements of the set S. Assume that S = {.

**Input Format:**

The file contains lines. The first line contains a positive integer which denotes the size of the matrix. The next rows and columns represent the 0-1 matrix. The elements in each row are separated by a space.

**Output Format:** A single string composed of elements of S (in a topological order) such that every two elements are separated by a $.

Ex: For n=6, if the topological sorting is 1 3 5 4 6 2, then your output must be of the form: 1$3$5$4$6$2.

1. Given a relation R, using 0-1 matrix, verify whether it is anti-symmetric and transitive or not.

**Input Format:**

The file contains lines. The first line contains a positive integer which denotes the size of the matrix. The next rows and columns represent the 0-1 matrix. The elements in each row are separated by a space.

**Output Format:**

If the relation is anti-symmetric, then output 1 else 0. Similarly, if the relation is transitive output 1 else print 0.

Your final output must be a two-digit binary number. The first digit is the answer for anti-symmetric and the second digit is the output for transitive property.

1. Find the transitive closure of a relation R, given in 0-1 Matrix.

**Input Format:**

The file contains lines. The first line contains a positive integer which denotes the size of the matrix. The next rows and columns represent the 0-1 matrix. The elements in each row are separated by a space.

**Output Format:** Transitive closure for the given relation, in 0-1 matrix, is also a 0- matrix. As in the problem 5, here also print a single string composed of the elements in the transitive closure such that

* 1. The two consecutive rows are separated by two $s ($$) and
  2. The two consecutive elements in a row are separated by a single $.