National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Ms. Syeda Tayyaba
Lab Instructor(s)	Samia Akhter Sonia Anum
Section	BDS-3B
Date	10-October-2022
Semester	Fall 2022

Topics:
Recursion & Merge sort

Practice Questions:

 Merge sort
Problem 1
You are the king of Pensville where you have 2N workers.All workers will be grouped in association of size 2,so a total of N associations have to be formed.The building speed of the
Ith worker is Ai .To make an association, you pick up 2 workers. Let the minimum building speed between both workers be x, then the association has the resultant building speed x.
You have to print the maximum value possible of the sum of building speeds of N associations if you make the associations optimally.
Constraints

$1 \leq N \leq 5 * 104$

$1 \leq Ai \leq 104$

Input

First line contains an integer *N*, representing the number of associations to be made.Next line contains 2N

space separated integers, denoting the building speeds of 2N workers.

Output

Print the maximum value possible of the sum of building speeds of all the associations.

Sample Input

2

1 3 1 2

Sample Output

3

Problem 2

Today is the 25th anniversary of Berland International School in Berland. On this auspicious Occasion, our friend Monk has been given the responsibility of preparing the Inventory for his school.There are exactly *N* teachers and *M* students in the school. Each of these teachers teaches arbitary number of students. However, each student is taught by exactly one teacher.Monk has been given the task of finding out for each teacher the Students he/she teaches. For each student Monk has been given the Student's Name and Age. However , Monk is too busy , So he has assigned this task to us.We need to print the name of the Teacher and the Name and age of the students that he/she teaches.However, there is a catch here. We need to print the list of students of each Teacher in Lexicographical order of their names . That is list of the teacher with lexicographically smaller name will appear before other teachers with lexicographically greater names.In Addition , The students appearing in a particular teachers list should appear in Increasing order of their Age.

Input Format :

The first line contains two integers *N* and *M* denoting the number of Teachers and number of Students respectively. Each of the next *N* lines contain a single string denoting a Teachers name.It is guaranteed that each teachers name shall be unique. The next *M* lines contain 2 Strings and an Integer, denoting the Teachers name, a Student's name being taught by that Teacher and that Student's Age. It is guaranteed that each Student's name shall be unique and shall appear only once in the Input.

Output Format:

Print N+M lines . Print the teachers name first and then the name and age of each student taught by this teacher. The list of each teacher should appear in order of their lexicographical rank in comparision to all other teachers. For example the list of the teacher with lexicographically smallest name should appear first, then the list of the teacher with the 2nd smallest lexicographical name and so on. The students in a particular teachers list should appear in the output in Increasing order of their Age.

Constraints:

$1 \leq N \leq 100$

$1 \leq M \leq 105$

$1 \leq$ Age of Eac Student $\leq 10power6$

$1 \leq$ Length of each Student and Teachers Name ≤ 35

The Name of Each Teacher and Student will consist of Lowercase English Alphabets only.It is guaranteed that no two students with the same age shall appear in the same Teacher's List.

Hint : You Need to Use Comparable Interface Here.

Sample Input

3 4

vasya

petya

kolya

vasya errichto 21

kolya petr 22

petya egor 19

vasya tourist 19

Sample Output

kolya

petr 22

petya

egor 19

vasya

tourist 19

errichto 21

Recursion
Problem statement:

You are given *N* eggs, and building with *K* floors from *1* to *K*. Each egg is identical, you drop an egg and if an egg breaks, you cannot drop it again. You know that there exists a floor *F* with *0* $<= F <= K$ such that any egg dropped at a floor higher than *F* will break, and any egg dropped at or below floor *F* will not break. Each move, you may take an egg (if you have an unbroken one) and drop it from any floor *X* (with *1* $<= X <= K$). Your goal is to know with certainty what the value of *F* is.

What is the minimum number of moves that you need to know with certainty what *F* is, regardless of the initial value of *F*?

Input:

The first line of the input is **T** denoting the number of test cases. Then **T** test cases follow. Each test case contains one line denoting *N* number of eggs and **K** denoting **K** number of floors.

Output:

For each test case output in a new line the minimum number of attempts that you would take.
F(F>=1 and F<=k)