CSE DEPT

Programming Contest 2006

Welcome to the Programming Contest

You are here to match your skills against other programmers and eventually be the best. Before we start, some general information about the contest.

The Contest

- You will have 4 questions to attempt in 2 hours
  - ① Choose the easiest questions first
  - ① Finish a question before attempting another one
- Each participant will have access to a computer with the standard tools for compiling and running Java and C++ as well as Eclipse on Linux
- The following documentation are available:
  - ① The Java 5 Documentation:
    - ■ http://cse.uom.ac.mu/java/index.html
  - ① The Standard Template Library Programmer's Guide
    - ■ http://cse.uom.ac.mu/cpp/index.html
- You are authorized to use any other source of documentation (Internet) that you think will be useful. However, don't waste time.
- No communication of any sort (talking, messages over the network,...) is allowed while the contest is on

Winner(s)

The winner is the one who does the maximum number of questions right in the minimum lapse of time.

Good Luck

You've already proven your worth just by being here!

The others have already chickened out ;)

Problem 1

The Jumping Champion

Professor Ma.L. (Math Lover) loves everything related to prime numbers. Remember that *a prime is a positive number bigger than one and only divisible by 1 and itself*. He is now working on a property of a set of primes called the jumping champion. An integer *N* is called the **"jumping champion"** if it is the most frequently occurring difference between consecutive primes.

For example, consider the consecutive primes 2 3 5 7 11. The differences between primes are 1 2 2 4. Therefore, for this set of primes, the jumping champion is exactly 2 (occurring two times).
He would really like to know for any set of primes what is their corresponding jumping champion. Could you help him?

Problem

Your task is to write a program that, given a lower and an upper bound, calculates the jumping champion of all the primes numbers that are in the defined limits (the upper and lower bound are considered themselves to be inside the limit).

Input

The first line of input contains an integer **T** which is the number of test cases that follow.
Each test case is given on a line with two integer numbers **L** and **U**, separated by a single space, which represent the lower and upper limits (respectively) to consider.

Output

The output consists of **T** lines, one for each case.
The i^{th} line contains:

- "The jumping champion is NUM" - if the jumping champion for the i^{th} case can be found and it is NUM ; if there is more than one difference occurring a maximum number of times, then anyone is chosen as the jumping champion.

Sample Input	Output for sample Input
3	
2 11	The jumping champion is 2
2 5	The jumping champion is 1 (Here, the jumping champion can also be 2)
30 50	The jumping champion is 4

Problem 2

Containers

Containers are to be filled with objects to be shipped. All containers are of the same width and length, although they can be of four different heights. Objects are of the same width as the containers, and can be of different lengths (smaller than the length of the containers) and of four different heights corresponding to the heights of containers.

Notes:

- We assume that if an object has the same dimensions as a container, it will fit the container.
- The width of all objects is always less than their height and length, such that objects can only be inserted along their length.
- Objects cannot be stacked one on top of the other.

Given that the maximum number of containers is 20, and the containers can be of any of the four heights, we are to optimize the number of objects sent as follows: The objects which are to be sent are to be categorized by height first, then by their length, such that the taller objects are sent first, then the longer objects. The tallest containers are filled with objects of the corresponding height first, starting by the longer objects. Once all objects of a certain height are exhausted, shorter objects will be considered to be inserted.

Notes:

- We are to use the minimum number of containers
- The tallest containers are to be used first

Input

The input will consist of multiple lines. The first line is the number of containers (NC) followed by NC lines describing each container, by giving the container number, its width, its length, and height.
Next we have the number of objects (NO) followed by NO lines describing each object (object number, its width, length and height).

Note: All dimensions are assumed to be in feet.

Output

For each container, list all the objects that fit it. Give the container number on a separate line, and the list of object numbers. Order the containers in descending order of height. If a container has no objects, just list its number.

Sample Input	Output for sample Input
6 1 10 100 100 2 10 100 75 3 10 100 25 4 10 100 75 5 10 100 50 6 10 100 100 7 1 10 25 50 2 10 50 100 3 10 25 25 4 10 75 100 5 10 25 75 6 10 50 75 7 10 50 50	1 4 5 6 2 7 2 6 1 4 3 5 3

Problem 3

Selection of Options

A car designer has to decide which set of options to offer in her new car. Naturally, she wants to select the largest number of options to increase the desirability of her car. But, as each option has a cost and a weight, she must choose a set of options that has:

- a combined cost which is less or equal a certain maximum cost and
- a combined weight which is less or equal a certain maximum weight.

She has at most 20 different options to choose from.

Input

Input consists of a number of test cases, each describing one scenario (that is a set of options, a maximum cost and a maximum weight)

Each case begins with a line containing NOPT, the number of distinct options (at most 20), MAXCOST, the maximum cost allowed and MAXWEIGHT, the maximum weight allowed.

Then follows NOPT lines, each containing the cost and weight of a particular option. The options are numbered 0 to NOPT-1 in the order they appear (i.e. the first option will be option 0 and the last one in the input will be option NOPT-1)

Both the cost and weight of each options are integers less than 10,000.

The last case is followed by a line containing a 0 0 0.

Output

For each test case, output one line containing the combined cost, the combined weight and the set of options (in increasing order) that satisfy the requirements in the format shown below.

We will assume that each solution is unique.

Sample Input	Output for sample Input
5 6000 1200 3000 400 2500 1000 2000 500 600 200 5000 100 6 10000 2000 5000 500 1000 750 2000 800 1800 1200 3000 50 500 650 0 0 0	Cost: 5600 Weight: 1100 Options: 0 2 3 Cost: 9500 Weight: 1950 Options: 0 1 4 5

Problem 4

The dating Agency

Kate is the proud manager of the very successful Love@firstSight dating agency. Lonely people submit their names to Kate who forms couples and inform the latter after about a week or so. The underlying success of Kate's agency is numerology. Every week, she pairs couples together based the following rule:

- For each pair of names, she removes all the duplicate characters (even in the same name)
  - e.g. fiona and bill: result: f, o, n, a, b (i – is common to fiona and bill, l is present twice in bill)
- then she sums up the remaining letters according to the following:
  - a: 1, b: 2, c: 3, d: 4, ... ,y: 25, z: 26
- the individual digits of the sum are repeatedly added with each other until there is only one number left
  - e.g. letters: f, o, n, a, b = 6 + 15 + 14 + 1 + 2 = 38 = 3 + 8 = 11 = 1+ 1 = 2
- the couple with the highest one digit sum is paired, following by the next highest,... until no possible couple is left. (In case, 2 numbers are same, anyone is chosen)

Everything was fine since one Wednesday, she fell sick and could not do the matching – Kate's astrologist told her to do the matching only on Wednesdays, else the newly formed couples will carry bad luck. So she thought of a program to help her solve that but unfortunately she's not skilled enough.
Moreover, her skills are so well known that other matching agencies ask her to help when they don't know how to pair their clients.

Input

Each line starts with 2 integers (less than 10 each) representing the number of males and females awaiting processing. The next line contains the first name of each male candidate separated by a space followed by another line containing the first name of the female candidates separated by a space as well. The last problem instance is followed by 0 0.

Output

Each line shows the possible couples for each agency.

Sample Input	Output for Sample Input
4 4 joe peter josh bill jane mary anita fiona 3 2 gary phil paul leslie lizzie 0 0	Couples: joe-fiona bill-anita josh-jane peter-mary Couples: paul-lizzie gary-leslie