06th June, 2009

NOTICE TO ALL CONTESTANTS OF THE IOI 2009 SELECTION TEST 1

Important Instructions:

You are advised to **strictly adhere** to the instructions given below. Those who do not comply with these will be subjected to reduction of their scores or disqualification.

1. The test consists of programming solutions to 3 problems. The duration of the test is 5 hours. You will receive regular announcements of the time remaining.

2. Do not take any material inside the lab without the permission of the Supervisor. Drinks/Snacks are not permitted inside the Lab but are provided in the reception area.

3. There will be paper for rough work near your machine. The keyboard will be placed on top of the monitor and the problems will be under the keyboard. **Do not touch the computer or keyboard until we announce START.**

4. You are allowed to ask written questions to clarify the problem statement(s) during the first 1 hour only. Answers will be limited to **Yes**, **No** or **No Comment**. Please phrase your questions accordingly.

5. At the end of the 5-hour period we will announce **STOP. Please stop work immediately and place your keyboard on top of the monitor.** The only exception to this is if you have been awarded additional time to compensate for machine failure.

6. There will be an overview sheet detailing the naming of program files, directories where they should be stored, input and output files, etc. with the problem statements. It will also contain details regarding number of test cases, point distribution and time limits per test case. **NOTE: Input/Output is file-based for this selection test.**

7. Your programs must be stored in the directory specified in the overview, compiled to EXE format. **NOTE: No online submissions for this selection test.**

8. Problem statements will include one or more examples. The fact that your program(s) works for the given example(s) does not guarantee that your program is correct.

9. All input data sets used for testing will conform to input specifications and will have valid outputs; however it is possible that test data generated by you might not have solutions.

10. The program must solve the test data within the given time limit. If the program exceeds the time limit for a test case, the output for that test case will not be evaluated.

11. Adhere strictly to the output format specified.

12. Save your work frequently in the working directory to avoid loss due to power failure and other unforeseen circumstances.

13. Technical support regarding use of the IDE, coding, debugging or testing will not be provided.

Any specific inquiries regarding results will be entertained only until 10th June, 2009.

IOI 2009 SELECTION TEST 1

6^{th} June, 2009

Duration: 5 hours

Task Name	File Name (*) to save your program	Compiled File name	Directory to save your program	Input	Output	Time Limit per Test	Number of Test Cases	Points per Test Case
DELIVERY	delivery.cpp delivery.c	DELIVERY.EXE	C:\ioi\delivery	DELIVERY.IN	DELIVERY.OUT	1 second	10	10
COINS	coins.cpp coins.c	COINS.EXE	C:\ioi\coins	COINS.IN	COINS.OUT	1 second	10	10
WRESTLE	wrestle.cpp wrestle.c	WRESTLE.EXE	C:\ioi\wrestle	WRESTLE.IN	WRESTLE.OUT	1 second	10	10

(*) Choose the source filename according to the programming language used.

DELIVERY

There is a large amount of packets that need to be delivered within a certain time limit if possible. There are two vehicles available for making the deliveries, a small van and a large lorry.

Each packet is either small or large. A small packet can be delivered either with the van or with the lorry, but a big one requires the lorry. For each packet we know the amount of time it would take to deliver it; we call this the time requirement of the packet. One vehicle can deliver only one packet at a time.

We are also given a deadline, which is the total amount of time within which we should deliver as many packets as possible. We do not care about what happens after the deadline.

The task in this problem is to determine the largest total amount of packets that can be delivered within the deadline.

CONSTRAINTS

$1 \le T \le 1000$, the deadline
$1 \le N \le 500$, the total number of small packets
$1 \le M \le 500$, the total number of large packets
1 ≤ time requirement to deliver a packet ≤ 1000

INPUT

The first input line contains a single integer T - the deadline.

The second input line contains a single integer N - the total number of small packets.

The next N input lines each contain a single integer - the time requirements of the small packets, in ascending order.

The next input line number contains a single integer M - the total number of large packets.

The next M input lines each contain a single integer - the time requirements of the large packets, also in ascending order.

OUTPUT

The only line of the output file should contain a single integer - the largest number of packets that can be delivered within the deadline.

EXAMPLE

Sample Input

```
10
8
2
2
2
2
2
4
4
4
4
3
3
6
6
```

Sample Output

```
8
```

The deadline is 10 time units. There are 8 small packets: 5 with time requirement 2 and 3 with time requirement 4. There are 4 large packets: 2 with time requirement 3 and 2 with time requirement 6.

The 8 packets can be delivered in 10 time units by delivering the small packets with time requirement 2 with the van, and using the lorry to deliver one small packet with time requirement 4 and 2 large packets with time requirement 3.

COINS

In a certain country, there are N denominations of coins in circulation, including the 1 cent coin. Additionally, there's a bill whose value of K cents is known to exceed any of the coins. There's a coin collector who wants to collect a specimen of each denomination of coins. He already has a few coins at home, but currently he only carries one K-cent bill in his wallet.

He's in a shop where there are items sold at all prices less than K cents (1 cent, 2 cents, 3 cents, … , $K-1$ cents). In this shop, the change is given using the following algorithm:

1. Let the amount of change to given be A-cents.
2. Find the highest denomination that does not exceed A. (Say it's the B-cent coin.)
3. Give the customer a B-cent coin and reduce A by B.
4. If $A = 0$, then end; otherwise return to step 2.

The coin collector buys one item, paying with his K-cent bill.

Your task is to write a program that determines:

1. How many different coins that the collector does not yet have in his collection can he acquire with this transaction?
2. What is the most expensive item the store can sell him in the process?

CONSTRAINTS

$1 \le N \le 500\,000$, the number of coins in circulation
$2 \le K \le 1\,000\,000\,000$, the denomination of the bill

INPUT

The first line of the input file contains the space separated integers N and K.

The following N lines describe the coins in circulation. The $(i+1)^{th}$ line contains the space separated integers c_i ($1 \le c_i < K$) and d_i, where c_i is the value (in cents) of the coin, and d_i is 1, if the collector already has this coin, or 0, if he does not.

The coins are given in the increasing order of values, that is, $c_1 < c_2 < \ldots < c_N$. The first coin (c_1) is the 1 cent coin (c_1=1).

OUTPUT

The first line of the output file should contain a single integer — the maximal number of denominations that the collector does not yet have, but could acquire with a single purchase.

The second line of the output file should also contain a single integer — the maximal price of the item to buy so that the change given would include the maximal number of new denominations, as declared on the first line.

EXAMPLE

Sample Input

```
7 25
1 0
2 0
3 1
5 0
10 0
13 0
20 0
```

Sample Output

```
3
6
```

Buying an item priced at 6 cents will result in a change of 25-6=19 cents, which will be given as 13, 5 and 1 cent coins. Note that there is no item more expensive than 6 cents that will give the collector 3 coins he doesn’t already have in his collection.

WRESTLE

Once upon a time, there lived a king who was extremely fond of wrestling. In those days, wrestlers had supernatural powers. To win a match a wrestler relied not only on his own strength but also on a magical ring that he wore while fighting. This ring allowed a wrestler to gain additional strength proportional to the strength of his opponent.

The strength of a wrestler and the magical power of his ring are both positive integers. When wrestler A fights wrestler B, the fight index of wrestler A for that match is given by A's own strength plus the strength of wrestler B multiplied by the magical power of A's ring. Each match is won by the wrestler whose fight index is higher for that match.

For example, suppose that wrestler A has strength 10 and wears a ring whose magical power is 3 and wrestler B has strength 18 and wears a ring whose magical power is 4. If these two wrestlers fight each other, wrestler A wins. This is because A's fight index for this match is $10 + (3 \times 18) = 64$ while B's fight index for this match is only $18 + (10 \times 4) = 58$.

On the other hand if A faces a wrestler C with strength 15 and a ring whose magical power is 5, then C wins. In this match, A's fight index is $10 + (3 \times 15) = 55$ while C's fight index is $15 + (5 \times 10) = 65$. Similarly, in a match between B and C, C wins.

The king organised a wrestling festival once a year. During this festival, each wrestler fought every other wrestler exactly once. At the end of the contest king honoured all the wrestlers by inviting them to his court and giving them gold coins.

It was the job of the Minister to decide the order in which the wrestlers got to meet the king. This was an important task, because the king was rather eccentric. He had declared that the number of coins to be given to a wrestler was determined by the number of matches he won and his position in the sequence: a wrestler was given one gold coin for each match he won and one gold coin for each wrestler whom he defeated but who was ahead of him in the line to meet the king.

For instance if the Minister had presented the wrestlers A, B and C described above in the order A,B,C to the king, then A would get 1 gold coin (for the match he won against B), B would get 0 gold coins (since he won no matches) and C would get 4 gold coins (two for his wins against A and B, one because he beat A and A met the king before him and one more since he beat B and B met the king before him). Instead if the minister had presented them in the order C, A, B then C would get 2 coins (for his wins against A and B), A would get 1 coin (for his win against B) and B would get 0 coins.

The Minister is concerned about the finances of the kingdom and wants to minimise the number of coins handed out. Your task is to help the Minister decide the sequence in which the wrestlers should be presented so that the number of gold coins handed out is minimized.

You are provided with the strengths of all the wrestlers and the magical powers of their rings. You may assume that these values are such that no match results in a tie.

CONSTRAINTS

$1 \le N \le 10\,000$, the number of wrestlers
$1 \le S_i \le 1\,000$, the strength of the i^{th} wrestler
$1 \le R_i \le 1\,000$, the magical power of the i^{th} wrestler's ring

INPUT

The first line of input is an integer N indicating the number of wrestlers.

Each wrestler is identified by a unique number in the range {1, 2, . . . ,N}. The next N lines each contain 2 positive space separated integers. For $1 \le i \le N$, the first integer on line i+1 denotes the strength S_i of wrestler i and the second integer denotes the magical power R_i of wrestler i's ring

OUTPUT

The output should consist of N lines, indicating the order in which the wrestlers should meet the king.

Thus, each line of output should be an integer between 1 and N and every integer between 1 and N should appear exactly once in the output.

If the integer on line i is j, it means that wrestler j is the i^{th} wrestler to meet the king.

EXAMPLE

Sample Input

```
3
10 3
18 4
15 5
```

Sample Output

```
3
1
2
```