Sapientia Hungarian University of Transylvania

ECN International Programming Contest 2017

Problems

Târgu Mureş / Marosvásárhely, 13th May 2017

Problem A

Input file :	**standard input**
Output file :	**standard output**

The Ministry of Healthcare must implement a software system for the management of expenses. The input contains:

- on the first line, an integer ($n \leq 200000$), the number of persons having health insurance; they don't pay any medical service, so every requirement is recorded together with its price;
- on the next n lines, the personal code (31 bits integer) and the name (27 characters or less including space).

After the last person, the history of requirements follows. Every line is related to one requirement:

- the personal code of the person who required medical service; for statistical purposes, the persons without health insurance are also recorded, but they pay every medical service;
- if the person is insured, the personal code is always followed by the price of medical service (31 bits integer);
- if the person is not insured and the price is present, you simply ignore the price.

The maximum length of every history line is 27 characters. Using the template below, you have to display:

- the total value of medical services, ignoring those which are related to persons without health insurance;
- the personal code and the name of persons who required the maximum number of medical services, together with this maximum number;
- the personal code and the name of persons who cumulated the maximum value of medical services, together with this maximum price;
- every personal code must be preceded by one space, every list must be ordered alphabetically;
- the number of persons without health insurance who required medical services.

Every two consecutive sections must be separated by an empty line. Only the total value (first section) may exceed 32 bits integer.

standard input	standard output
5 1 Arthur Conan Doyle 2 Edgar Alan Poe 3 George Sand 4 Marie Curie 5 Mireille Mathieu 1 6 2 7 3 10 1 6 2 6 3 10 1 6 2 7 4 18 6 7 5 8 9 6 4 7 8 6 3 7	Total value of medical services: 76 Maximum number of medical services: 3 1 Arthur Conan Doyle 2 Edgar Alan Poe Maximum cumulated value: 20 2 Edgar Alan Poe 3 George Sand Persons without insurance: 4

Problem B

Input file :	**standard input**
Output file :	**standard output**

There are a lot of people on the Tg. Mure stock market. Some buy shares, others sell. Transactions are recorded electronically at Smart Software Company, where programmers are prepared to answer various questions. This morning, the business manager asked something from Rookie, a recently hired programmer, but he could not answer. If you succeed to answer the question, the manager will give you an award and who knows he might even hire you at the Smart Software Company. Managers question: What is the value of the transaction which is larger (or equal) than $k-1$ other transactions?

Input: The first line of the input contains the number of test cases, t. The first line of each test case includes the number n (the total number of transactions in the Smart database) and the number k (with the meaning given in the requirement). Each of the following n lines contains an integer (positive numbers are incomes for the market, negative numbers represent payments). Two test cases are separated by an empty line.

Output: For each test case your program has to print on a separate line the value of the traded amount that is the k^{th} smallest value in the Smart database.

Restrictions and refinements

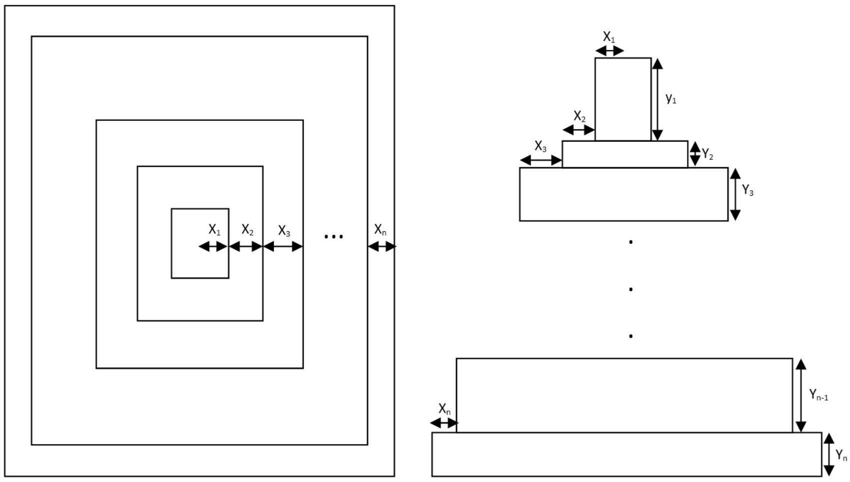
- $1 \le n \le 10^6$
- $1 \le k \le n$
- $-2^{31} - 1 \le$ traded amount $\le 2^{31} - 1$
- Numbers are not necessarily distinct
- The smallest amount has the order number 1

Standard input	Standard output	
1 10 8 2356700 -45001 10023 450 -19999023 876 -76005 876 3299881 -451	10023	The numbers in ascending order: 1. -19999023 2. -76005 3. -45001 4. -451 5. 450 6. 876 7. 876 8. 10023 9. 2356700 10. 3299881

Problem C

Input file :	**standard input**
Output file :	**standard output**

The key parameters of the "gradual square pyramid" from the figures are stair-depth (x) and stair-height (y). Given the $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$ pair-sequence ($n \leq 20,\ x_i < 100,\ y_i < 10,\ i = \overline{1, n}$), find out the optimal succession that minimize the volume of the pyramid.



Input: The first line of the input file contains the number of the given test cases: the value of n followed by pairs of integers representing the stair-height and -depth separated with white characters: $y_1\ x_1\ y_2\ x_2 \ldots y_n\ x_n$.

Output: A line for each test case written to the standard output: the indexes of the height-depth pairs in the correct order separated with white characters.

Standard input	Standard output
1 6 3 10 2 11 1 5 3 11 2 5 1 10	5 1 4 3 2 6

Problem D

Input file :	**standard input**
Output file :	**standard output**

The problem: is described in the input.

Problem E

Input file :	**standard input**
Output file :	**standard output**

For the first edition of the Sapientia-ECN contest, continuous bus transport was organized from the train station (A) to Sapientia University (B) and back. The bus line from A to B is divided into N unit distance (u_D) parts (so the total length of the route is $N \cdot u_D$). Any given bus travels a unit distance in a unit time (u_T). Departure times from the train station (A) are at times $0 \cdot u_T$, $a \cdot u_T$, $2 \cdot a \cdot u_T, \ldots$ (over the years the bus traffic between the train station and Sapientia University has become a permanent and continuous arrangement, so busses will depart continuously at moments multiple of $a \cdot u_T$). When a bus arrives to the University (B) it will wait till the closest time spot which is an integer multiple of $a \cdot u_T$ and then it returns to the train station (A). Along the route $A \rightarrow B$ there are stops at positions $0 \cdot u_D$, $x_1 \cdot u_D$, $x_2 \cdot u_D$,..., $x_n \cdot u_D$, $N \cdot u_D$, and on the way back, $B \rightarrow A$, at positions $N \cdot u_D$, $y_n \cdot u_D$, $y_{n-1} \cdot u_D$,..., $y_1 \cdot u_D$, $0 \cdot u_D$ of the bus line. At every stop the bus waits $b \cdot u_T$ time.

Which are those u_D multiple points ($[i \cdot u_D], i \in \{0, 1, \ldots, N\}$) and u_D length open intervals ($(i \cdot u_D, (i+1) \cdot u_D)$, $i \in \{0, 1, \ldots, N-1\}$) of the bus line, where two buses will meet? Only the meeting point indexes ($[i]$, $i \in \{0, 1, \ldots, N\}$) and the endpoints indexes of the meeting intervals ($(i, (i+1))$, $i \in \{0, 1, \ldots, N-1\}$) have to be printed.

Input: format for each test cases
u_T u_D
N n
a b
x_1 $x_2 \ldots x_n$
y_1 $y_2 \cdots y_n$

Output: for each test case you have to print in consecutive lines the u_T multiple meeting point indexes (in square brackets) and meeting interval endpoints indexes (in brackets, separated with comma) (sorted in ascending order) followed by the number of meetings.

Restrictions: All input values are integers:

- $0 < N < 1000$
- $0 < u_T, u_D < 100$
- $0 < a, b < 50$
- $0 < x_1 < x_2 < ... < x_n < N$
- $0 < y_1 < y_2 < ... < y_n < N$

Standard input the first line contains the number of test cases	Standard output
1 3 11 100 5 20 3 20 40 60 70 90 20 40 50 80 90	[0] (7,8) (17,18) (24,25) (34,35) (41,42) [50] [60] (68,69) [77] (85,86) (92,93) [100] 13

Problem F

Input file :	**-**
Output file :	**standard output**

It is given a random number generator implemented in C programming language.

```
unsigned int m_w = 11;
unsigned int m_z = 173;
unsigned int M = 2001;


unsigned int myrandom() {
      m_z = 36969 * (m_z & 65535) + (m_z >> 16);
      m_w = 18000 * (m_w & 65535) + (m_w >> 16);
      return ((m_z << 16) + m_w) % M;
}
```

- A & B bitwise AND operator
- A >>n - bitwise right shift operator
- A <<n - bitwise left shift operator
- A % M - modulo operator
- A = B - assignment

Using the given random number generator generate $N = 100000$ distinct points having integer coordinates between 0 and 2000 in the plane. The numbers are generated until the desired number of distinct points is reached. Compute the shortest distance between the generated points.

A point in the plane is generated by two consecutive call of `myrandom()` function.

```
unsigned int x = myrandom();
unsigned int y = myrandom();
```

Input: No input is required.

Output: The shortest distance, followed by a newline character.

Problem G

Input file :	**standard input**
Output file :	**standard output**

For any positive integer x the classical Möbius function $\mu(x)$ has values in $\{-1, 0, 1\}$ defined in the following way:

$$\mu(x) = \begin{cases} 1 & \text{if } x \text{ is a square-free positive integer with an even number of prime factors} \\ -1 & \text{if } x \text{ is a square-free positive integer with an odd number of prime factors} \\ 0 & \text{if } x \text{ has a squared prime factor} \end{cases}$$

Write a program that for a positive integer calculates the value of Möbius function.

The input contains several numbers where each number is less than 2 000 000 000. For each numbers calculates the corresponding value of Möbius function and write the result on standard output, each value in separate line.

Standard input	Standard output
3	-1
4	0
17	-1
48	0
2 727	0
605 700 457	1
1 092 678 973	-1
1 999 999 999	1

Problem H

Input file :	**standard input**
Output file :	**standard output**

Given an array of N numbers, we wish to choose a contiguous subsequence of the array, so that the bitwise XOR of all chosen numbers is maximum. Bitwise XOR is defined as follows: every bit in the answer is obtained by applying XOR logic on the corresponding bits of the set of numbers:
0 XOR 0 = 0
0 XOR 1 = 1
1 XOR 0 = 1
1 XOR 1 = 0

For example 7, 8 and 5 are XOR-ed as follows, Numbers in binary:
0111
1000
0101
$\vdots$
1010

So the answer is 10 (in decimal). The same answer can be obtained in C,C++, Java by using the XOR operator as 7ˆ8ˆ5.
Input: The first line contains one integer, N, the size of the array. The next line contains integers denoting the elements of the array, separated by a single space.

Output: Output a single line containing the maximum sum that can be obtained.

Restrictions:

- $1 \leq N \leq 100.000$
- $1 \leq$ any element of the array $< 2^{21}(2.097.152)$

Standard input	Standard output
5 3 8 2 6 4	15

Problem I

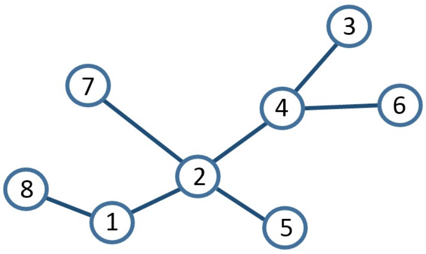
Input file :	**standard input**
Output file :	**standard output**

In some far away region, there are N cities, connected by $N-1$ bidirectional roads so that between any two cities there is one unique road (one direct road or one formed by several direct roads). For any city i it is known the population P_i of the city.

One of the N cities will become the regions cultural capital. This will be decided by the following properties:

- The capital has to be directly connected by at least two other cities.
- In order to stabilize the traffic on the roads entering the capital; the capital has to be the city for which the traffic on the incident edges has the smallest variation. The traffic on one edge is given by the total population which uses that edge to go to the capital. The variation is the maximum absolute difference between the values.

For example:



i	1	2	3	4	5	6	7	8
P_i	2	5	1	2	5	2	6	4

City number 2 is the one to become the capital since it is directly connected by at least two other cities (4 in this case), and the population that will use the incident edges of city 2 is:
`edge (2,1) :` $P_1 + P_8 = 2 + 4 = 6$
`edge (2,4) :` $P_3 + P_4 + P_6 = 1 + 2 + 2 = 5$
`edge (2,5) :` $P_5 = 5$
`edge (2,7) :` $P_7 = 6$
The absolute *variation* is 6 5 = 1 and there is no other city with smaller variation. Given the region defined by cities, population and roads, find the city that will become capital, according to the requirements and also the traffic variation for that city.
Input: The input contains on the first line N, the number of cities. On the second line there are N numbers, the values P_i in order, separated by spaces. Next $N-1$ lines contain two numbers i and j that specify cities that are connected by direct roads.

Output: The output will contain only one line with two numbers separated by space: the index of the city that will become the capital and the smallest variation between the values of the traffic on the incident edges of the capital. If there are multiple cities that can become capital (with minimum variation), print the one with the smallest index.

Restrictions:

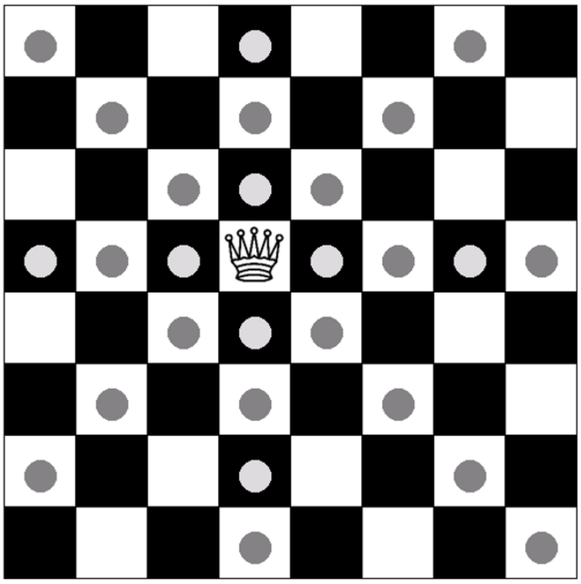
- $3 \leq N \leq 20000$
- $1 \leq P_i \leq 100000$
- Time limit per test: 0.6 sec

Standard input	Standard output
8 2 5 1 2 5 2 6 4 3 4 6 4 4 2 7 2 2 5 1 8 2 1	2 1

Problem J

Input file :	**standard input**
Output file :	**standard output**

On the chessboard, the queen is able to move any number of squares vertically, horizontally or diagonally (see the figure). Queen A captures queen B if queen A can move to the square of queen B in one step. There is an $N \times N$ chessboard and there are Q pieces of queens on it.



The task is to count how many groups the queens form. Queen A and queen B belong to the same group if they capture each other or there are queens q_1, q_2,...,q_k so that queen A captures queen q_1, queen q_i captures queen $q_{i+1}(1 \leq i < k)$ and queen q_k captures queen B.

Input: The first line of the input contains the number of test cases (t). The first line of each test case contains two positive integer numbers: N is the size of the chessboard and Q is the number of queens. The next Q lines contain the (x_i, y_i) coordinates of the queens ($0 \leq x_i, y_i \leq N-1, 1 \leq i \leq Q$). There are no two or more queens on the same square.

Output: For each test case, the program has to write out into a separate line the number of groups the queens form.

Restrictions:

- $1 \leq t \leq 100$
- $1 \leq N \leq 10^9$
- $1 \leq Q \leq \min\{N \times N, 1000\}$
- $0 \leq x_i, y_i \leq N-1$

Standard input	Standard output
2 8 5 0 0 2 4 5 4 0 2 6 3 8 3 2 1 3 6 7 5	1 3

Problem K

Input file :	**standard input**
Output file :	**standard output**

We have a number of monochrome unit cubes colored with n different colors (we have more of each color). Our target is to build with 8 randomly picked cubes as many differently colored $2 \times 2 \times 2$ cubes as possible. Two cubes are considered identical if we can get it through rotating another construction, otherwise they are considered different colorings.

Example: having two colors, i.e. black and white unit cubes we can build 23 differently colored $2 \times 2 \times 2$ cubes. Considering that there are enough cubes in n different colors, compute the number of differently colored $2 \times 2 \times 2$ cubes that can be built using them.

Input: On the first line of the input we have the number C of test cases. Each of the next C lines contains a number, the number of the different colors of the unit cubes.

Output: The output will contain C lines, every line will contain the number of differently colored cubes modulo 666013 that can be obtained with n colors. The order of the numbers will correspond to the input lines order.

Restrictions:

- $0 \le n \le 2000000000$

Standard input	Standard output	Explanation
1 2	23	We have a single case, use two colors e.g. black and white. Cases: - 1 case using 8 white unit cubes; - 1 case using 7 white unit cubes and 1 black unit cube; - 3 cases using 6 white unit cubes and 2 black unit cubes; - 3 cases using 5 white unit cubes and 3 black unit cubes; - 7 cases using 4 white unit cubes and 4 black unit cubes; - 3 cases using 3 white unit cubes and 5 black unit cubes; - 3 cases using 2 white unit cubes and 6 black unit cubes; - 1 case using 1 white unit cube and 7 black unit cubes; - 1 case using 8 black unit cubes. Total: 23 different colorings.
4 2 1000 5000 2000000000	23 45538 280943 185532	We have four cases with 2, 1000, 5000 and 2000000000 colors. Results modulo 666013 are: 23, 45538, 280943 and 185532.

Problem L

Input file :	**standard input**
Output file :	**standard output**

Nerdy Bernie takes notes diligently at the university lectures and adds page numbers to his pages, starting at 1, incrementing by 1. However, on his way home, stormy winds tear his notes out of his hands, scattering the pages around him. While he is gathering his notes, the rain begins to fall, soaking some of the pages. Of course, when Bernie arrives home, he immediately starts to lay them out to dry, but the pages with page numbers blurred by the rain are no longer usable or recoverable. Your job is to help Bernie determine those continuously numbered ranges of pages that were soaked wet in the rainstorm.

Input: The first line of the input contains two integers N and M, separated by exactly one space. N represents the number of numbered pages Bernie had before leaving the university for home ($0 < N < 2^{31}$), and M represents the number of pages with page numbers still readable ($0 \leq M \leq 1000$ and $M \leq N$). The next line consists of exactly M different integers a_i ($0 < a_i \leq N$, where $1 \leq i \leq M$), representing the page numbers of the pages with page numbers still readable. Consecutive numbers are separated by a space.

Output: The program should output the continuously numbered ranges of pages that were soaked wet, by printing the first and the last page number of each range, with a hyphen ('-' character) between them. Print the ranges in ascending order, one range per line. If a range contains only one page, the first and last page number should be the same.

Standard input	Standard output
10 3 1 4 9	2-3 5-8 10-10

Problem M

Input file :	**standard input**
Output file :	**standard output**

Let $X = (x_1, x_2, \ldots, x_n)$ be a sequence. A sequence $S = (s_1, s_2, \ldots, s_k)$ is a **subsequence** of X if there exists the integers $0 < i_1 < i_2 < \ldots i_k \leq n$ such that $s_j = x_{i_j}$ for $j = 1, 2, \ldots, k$. For example: 2, 5, 7, 3 is a subsequence of 2, 9, 5, 7, 8, 1, 3, 4.

Given three sequences of integers: $X = (x_1, x_2, \ldots, x_m)$, $Y = (y_1, y_2, \ldots, y_n)$, and $Z = (z_1, z_2, \ldots, z_k)$, find the longest common subsequence. For example, for the sequences $X = (2, 9, 5, 7, 8), Y = (9, 3, 5, 8), Z = (1, 9, 4, 5, 8)$ the longest common subsequence is (9, 5, 8).

Input

• The first line contains the number of tests.

For each test there are four lines:
• The first line of them contains m, n, k, the numbers of integers in each sequence, separated by a space.
• Each of the next three lines contains the elements (integer numbers) of a sequence, separated by a space.

Output

For each test, print in a line the elements of the longest common subsequence, separated by a space. If there is no solution, print the text: No solution

Restrictions and refinements

• $1 \leq m \leq 100$, $1 \leq n \leq 100$, $1 \leq k \leq 100$
• The sequence elements are integers from the interval $[-32768, 32767]$.

Standard input	Standard output
2 5 4 5 2 9 5 7 8 9 3 5 8 1 9 4 5 8 4 5 3 1 2 3 4 4 3 1 2 0 7 8 9	9 5 8 No solution