Gran Premio de México 2021 - Repechaje

December 18th, 2021

Problems book

General Information

This problem set contains 12 problems; pages are numbered from 1 to 18, Without considering this page. Please, verify your book is complete.

A) About your program

1) The code submitted to resolve a problem should be named : *problem_code*`.c`, *problem_code*`.cpp`, *problem_code*`.java` or *problem_code*`.py`, *problem_code*`.kt`, where *problem_code* is the uppercase letter that identifies the problem. Remember that in Java the name of the main class should be the same as the name of the file.

B) About Input

1) Your program should read the input from *standard input*.
2) When an input line contains more than one value, these values are separated by a single white space. The input does not contain any other white spaces.
3) Each line of input, including the last one contains exactly only one end of line character.
4) End of input is on the end of file.

C) About output

1) The output of your program should be written to *standard output*.
2) When an output line contains more than one value, these values should be separated by a white space. The output does not contain any other white space.
3) Each line of output, including the last one, should contain exactly one end of line character.

Problem A
A Code With Numbers

Adrian and Maria are relatives that live in different towns. As they inhabit a rural area, it is very difficult for them to keep in touch. One way they found to overcome their communication problem was to establish a line through their parents, who usually visit each other.

The point is that Adrian and Maria did not want that their parents to read their messages, so they decided to create a secret code for the messages. The code is not very sophisticated , but you should keep in mind Adrian and Maria are just children.

In general, the meaning of a message is based on changing some letters by numbers. Each message is composed by several lines using uppercase letters of the English alphabet, spaces and punctuation symbols: dot and comma. The letters that are changed by numbers can be seen in the following example: this change is the same for all messages between Adrian and Maria.

Message in "Code Number":

```
H3LL0 MY L0V3, 1 M H499Y 83C4U53 500N 1 W1LL 83 70 Y0UR 51D3.
7H15 71M3 W17H0U7 Y0U H45 833N 373RN4L. 1 1NV173 Y0U 70 7H3
200 0N3 70 533 7H3 238R42 4ND 60R1L42.
```

Decoded Message:

```
HELLO MY LOVE, I M HAPPY BECAUSE SOON I WILL BE TO YOUR SIDE.
THIS TIME WITHOUT YOU HAS BEEN ETERNAL. I INVITE YOU TO THE
ZOO ONE TO SEE THE ZEBRAS AND GORILAS.
```

Your task is to write a program to help Adrian and Maria to decode the messages they receive in Code Number.

Input

The first line of the input contains a single integer number T, representing the number of test cases. Each test case consists of a line with a string in "Code Number", where the words in the line are separated by one white space.

- $1 \leq T \leq 50$
- No line in the input will exceed 80 characters.

Output

For each test case in the input, print the decoded message.

Input example 1

```
3
H3LL0 MY L0V3, 1 M H499Y 83C4U53 500N 1 W1LL 83 70 Y0UR 51D3. 7H15
71M3 W17H0U7 Y0U H45 833N 373RN4L. 1 1NV173 Y0U 70 7H3 200 0N3 70
533 7H3 238R45 4ND 60R1L45.
```

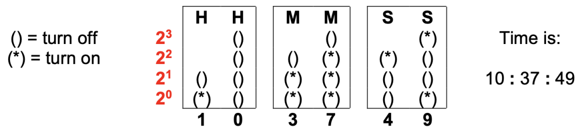
Output example 1

```
HELLO MY LOVE, I M HAPPY BECAUSE SOON I WILL BE TO YOUR SIDE. THIS
TIME WITHOUT YOU HAS BEEN ETERNAL. I INVITE YOU TO THE ZOO ONE TO
SEE THE ZEBRAS AND GORILAS.
```

Problem B
Binary Clock

A binary clock is a clock which displays traditional sexagesimal time in a binary format. The most common binary clock uses six columns of LEDs to represent zeros and ones. Each column represents a single decimal digit, a format known as binary-coded decimal (BCD). The bottom row in each column represents 1 (or 2^0), with each row above representing higher powers of two, up to 2^3 (or 8). To read each individual digit in the time, the user adds the values that each illuminated LED represents, and then reads these from left to right. The first two columns represent the hour, the next two represent the minute, and the last two represent the second.

For example:



Your task for this problem is simple: read an hour in its binary format and output its equivalent in sexagesimal time format.

In order to facilitate your task, each one of the six columns of LEDs than represents a single decimal digit is concatenated as shown below.

For example, 10:37:49 would be written as: ()(*)()()()()()(*)(*)()(*)(*)(*)(*)()()(*)()()(*)

Note: There are no spaces between characters.

Input

The input consists of multiple test cases. The first line of input contains a single integer T which is the number of test cases that follow. Each test case consists of a single line containing a string that represents an hour in its binary format.

- $1 \leq T \leq 1000$
- Each LEDs string is guaranteed to represent a valid time between 00:00:00 and 23:59:59.
- The length of each LEDs string is guaranteed to be ≤ 60.

Output

For each test case, print a line in the format "Case c: *time*" where c is the test case number (starting with 1), and *time* is the equivalent in sexagesimal time format (HH:MM:SS) for the binary time from the case.

Input example 1

```
3
()(*)()()()()()(*)(*)()(*)(*)(*)(*)()()(*)()()(*)
()()()()()()()()()()()()()()()()()()()()
(*)()()()(*)(*)()()()()()()()()()()()()()()
```

Output example 1

```
Case 1: 10:37:49
Case 2: 00:00:00
Case 3: 23:00:00
```

Problem C
Candy Box

Christmas is around the corner and Gatuno wants to get Marin a gift. Gatuno did a deep research to find out Marin's favorite types of candy, and finally found the store where that candy is sold. Such candy is sold in beautiful rectangular boxes that vary in size and include different types of candy.

But, to achieve the perfect gift, it is not enough to choose the correct sweets, it is also necessary to put them in the correct arrangement so that Marin does not get mad when opening the box. There are two basic rules to follow so that the gift is perfect. First, no two sweets of the same type must appear in the same row or column, like in Sudoku. The store sells the types of sweets randomly, and it is not hygienic to move the sweets around, so Gatuno decided to eliminate the sweets that do not meet this condition. And second, to be discreet with the sweets that were removed, there should not be two vertically or horizontally adjacent empty spaces in the box.

You want Marin to eat the most amount of candy possible, but if the candy box doesn't meet the above rules, the gift could be highly unpleasant for Marin. So, your task is to write a program that, given the initial conditions of the box of sweets and the sweets inside it, determines the least amount of sweets that must be eliminated from the box for the gift to be perfect. It is guaranteed that the box will always have at least one combination that allows it to be perfect.

Input

The first line contains three integers W, H, T ($2 \leq W, H, T \leq 8$), where W represents the width of the box, H represents the height, and T represents how many different types of candy are in this box. Next, there are H lines, each one with W letters, where each letter is an uppercase letter from the english alphabet. Each letter represents the type of candy in the box.

Output

The output must be a single integer, which is the minimum amount of candy that should be removed from the box to make it the perfect box for Marin.

Input example 1	**Output example 1**
3 2 3 AAA CCB	3

Input example 2	**Output example 2**
5 5 5 ABCDA EABDD DEABA CCEBB BCEEB	7

Input example 3	**Output example 3**

`4 4 4`
`ABCD`
`DABC`
`CDAB`
`BCDA`
	`0`

Problem D
5D Tic Tac Toe

We are all familiar with the game of tic-tac-toe which is played in a 3x3 board and each player takes alternate turns marking an empty cell with either 'X' or 'O'. The game is won by the first player to put their symbol across either a line or a diagonal in the board.

That game can be generalized to any dimension and any size of board. In this problem we consider a tic-tac-toe game played in a 5x5x5x5x5 hypercube. The players take alternate turns marking any unmarked cell with either 'X' or 'O'. The game is won by the first player to put their symbol across any set of 5 cells that lie on a straight line. For example, the set of points $(0,0,0,0,0)$, $(1,0,1,0,0)$, $(2,0,2,0,0)$, $(3,0,3,0,0)$, $(4,0,4,0,0)$ lies in a straight line. This is another example: $(0,2,1,4,3)$, $(1,2,1,3,3)$, $(2,2,1,2,3)$, $(3,2,1,1,3)$, $(4,3,1,0,3)$.

Your task is to read in the number of moves and the description of the moves made by the players and determine which of the two players won the game and at which turn.

Input

The first line contains an integer t representing the number of turns played in total.

Each of the next t lines contains 5 integers and one character, representing the coordinates of the cell played in the given turn and the character will be either 'X' or 'O' depending on whose turn it was.

- $1 \leq t \leq 3125$
- All coordinate values in the input will be either $0, 1, 2, 3,$ or 4.
- It is guaranteed that no cell will be repeated in the input.

Output

The output will be one line containing the character corresponding to the wining player along with the 1-based index of the turn at which they won the game. If at the end of the t turns neither player has won, write "DRAW".

Input example 1	Output example 1
8 0 0 0 0 0 X 1 2 1 3 3 O 1 0 1 0 0 X 0 2 1 4 3 O 2 0 2 0 0 X 2 2 1 2 3 O 3 0 3 0 0 X 3 2 1 1 3 O	DRAW

Input example 2	**Output example 2**

`10`
`0 0 0 0 0 X`
`1 2 1 3 3 O`
`1 0 1 0 0 X`
`0 2 1 4 3 O`
`2 0 2 0 0 X`
`2 2 1 2 3 O`
`3 0 3 0 0 X`
`3 2 1 1 3 O`
`4 0 4 0 0 X`
`4 3 1 0 3 O`
	`X 9`

Problem E

Egyptian Multiplication

Ancient Egyptian multiplication is a systematic method for multiplying two numbers (known as multiplier and multiplicand) that does not require the multiplication table, only the ability to multiply by 2, and to add. Also known as Egyptian multiplication and Peasant multiplication, it decomposes the multiplier into a sum of powers of two and creates a table of doublings for the multiplicand. This method may be called mediation and duplication, where mediation means halving one number and duplication means doubling the other number.

This method has three phases: the decomposition, the table and the result.

The decomposition of a number N thus consists of finding the powers of two of which the number N is made up. The Egyptians knew empirically that a given power of two would only appear once in a number. For the decomposition, they proceeded methodically; they would initially find the largest power of two less than or equal to the number in question, subtract it out and repeat until nothing remained. (The Egyptians did not make use of the number zero in mathematics).

Example of the decomposition of the number $N = 13$:

- The largest power of two less than or equal to 13 is 8, $13 - 8 = 5$,
- The largest power of two less than or equal to 5 is 4, $5 - 4 = 1$,
- The largest power of two less than or equal to 1 is 1, $1 - 1 = 0$

$N = 13$ is thus the sum of the powers of two: 8, 4 and 1.

After the decomposition of the multiplier (N), it is necessary to construct a table with the powers of two times the multiplicand (M), starting from one, up to the largest power of two found during the decomposition. In the table, a line is obtained by multiplying the preceding line by two.

For example, if the largest power of two found during the decomposition of $N = 13$ is 8 and $M = 238$, the table is created as follows:

N	**M**
*1	238
2	476
*4	952
*8	1904

Finally, the result is obtained by adding the numbers from the second column from each row for which the corresponding power of two makes up part of the decomposition of N (marked with a *).

Thus, the result of the multiplication of $13x238$ is obtained as the addition of: $1904 + 952 + 238 = 3094$ or $238 + 952 + 1904 = 3094$.

Input

The input consists of multiple test cases. Each test case consists of a single line containing two integers N and M that represent the multiplier and the multiplicand, and a begin addition specifier, that indicates whether the sum will be carried out taking the values of the rows from top to bottom or from bottom to top, in order to obtain the result. The begin addition specifier is one of the letters u or b, where u indicates that you will start taking the values from the top row to the bottom row, and b indicates that you will start taking the values from the bottown row to the top row.

The last test case is followed by -1 on a line by itself.

- $0 \leq N, M \leq 10^9$

Output

For each test case, print a string in the format "Case c: N x M = $result$" where $result$ is the string representing the sum that should me made to complete the egyptian multiplication. The numbers in $result$ should be separated by a space, a character '+', and a space. In the event that the values of N or M are 0, then the answer must be only 0 for that case.

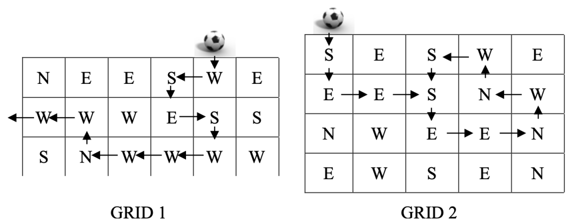
Input example 1

```
13 238 u
13 238 b
1000 1 u
1 1000 b
1 1 u
0 10 u
-1
```

Output example 1

```
Case 1: 13 x 238 = 238 + 952 + 1904
Case 2: 13 x 238 = 1904 + 952 + 238
Case 3: 1000 x 1 = 8 + 32 + 64 + 128 + 256 + 512
Case 4: 1 x 1000 = 1000
Case 5: 1 x 1 = 1
Case 6: 0 x 10 = 0
```

Problem F
Football



The football foundation (FOFO) has been researching on soccer; they created a set of sensors to describe the ball behavior based on a grid uniformly distributed on the field. They found that they could predict the ball movements based on historical analysis. Each square sensor of the grid can detect the following patterns:

- N north (up the field)
- S south (south the field)
- E east (to the right on the field)
- W west (to the left on the field)

For example, in grid 1, suppose the ball was thrown into the field from north side into the field. The path the sensors detected for this movement follows as shown. The ball went through 10 sensors before leaving the field.

Comparing with what happened on grid 2, the ball went through 3 sensors only once, and started a loop through 8 instructions and never exits the field.

You are selected to write a program in order to evaluate the ball behavior. The program needs to determine how many steps it takes for the ball to get out of the grid or, in case there is a loop, how many steps the ball takes before it enters such a loop and how many steps the loop has.

Input

There will be one or more grids of sensors for the same game. The data for each is in the following form. On the first line are three integers separated by blanks: The number of rows R in the grid, the number of columns C in the grid, and the number of the column E in which the ball enters from the north. The grid columns number starts with one at the left. Then come the rows of direction instructions. The lines of instructions contain only the characters 'N','S','E' or 'W' , with no blanks, nor spaces. The end of input is indicated by a grid containing 0 0 0 as limits.

- $1 \leq R, C \leq 20$
- $1 \leq E \leq C$

Output

For each grid in the input there is one line of output. Either the ball follows a certain number of sensors and exits the field on any one of the four sides or else the ball follows the behavior on some number of sensors repeatedly. The sample input below corresponds to the two grids above and illustrates the two forms of output. The word "step" is always immediately followed by "(s)" whether or not the number before is 1.

Input example 1	**Output example 1**

`3 6 5`
`NEESWE`
`WWWESS`
`SNWWWW`
`4 5 1`
`SESWE`
`EESNW`
`NWEEN`
`EWSEN`
`0 0 0`
	`10 step(s) to exit`
	`3 step(s) before a loop of 8 step(s)`

Problem G
GCD!

In mathematics, the greatest common divisor (GCD) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For two integers x, y, the greatest common divisor of x and y is denoted $gcd(x, y)$. For example, the GCD of 8 and 12 is 4, that is, $gcd(8, 12) = 4$.

The factorial of a non-negative integer n, denoted by $n!$, is the product of all positive integers less than or equal to n: $n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$ The value of $0! = 1$, according to the convention for an empty product.

Given two numbers x and y, your task is to find $gcd(x!, y)$.

Input

The first line of input contains a single integer T, representing the number of test cases. Each of the next T lines describe a test case with two integers separated by a space, x and y.

- $1 \leq T \leq 100$
- $0 \leq x \leq 2^{31}$
- $1 \leq y \leq 2^{31}$

Output

For each test case in the input print a line with a single integer, the value for $gcd(x!, y)$.

Input example 1	**Output example 1**
5 5 20 5 4 0 1000 1000 1 4 40	20 4 1 1 8

Problem H
How Exploitable

Working for one of the most prestigious information security companies, John's work is to analyse and seize the impact of vulnerabilities found on certain commonly used software libraries.

Today a vulnerability was found on a common string parsing library. After some research, John's team identified this vulnerability can be exploited only when the library parses a string S of length N that contains at least one of M different strings as substring. In order to seize the impact the vulnerability can have on internet systems, John's work is to determine how many different strings can be used to exploit the string parser.

John built a program to generate all the different strings, but, it is taking a lot of time to run, since this is a common use library, John needs to complete this task faster. Knowing your algorithmic skills, John asked for your help to complete the task. Can you help him find the number of different strings that can exploit the vulnerability faster?

Input

The first line of input contains two integer numbers separated by a space N, and M. Each of the next M lines contains a string, representing each of the strings that could be substring of S to exploit the vulnerability.

- $1 \leq N \leq 10^9$
- $1 \leq M \leq 10$
- Each of the M strings will contain only lowercase letters of the english alphabet, and will not exceed 10 characters.

Output

Print a line containing a single integer, the number of different strings S that can be used to exploit the vulnerability. As this number can be large print it modulo $10^9 + 7$.

Input example 1	Output example 1
`1 1` `a`	`1`

Input example 2	Output example 2

`2 2`
`ab`
`bb`
	`2`

Input example 3	Output example 3

`5 2`
`ab`
`bb`
	`136350`

Input example 4	Output example 4
`3 1` `aaaa`	`0`

Problem I
Baker's Cookies

Baker loves cookies (like most cats). His favorite are chocolate chip cookies, but he can't stand oatmeal cookies. Baker's human bought several boxes of cookies. Each box contains C cookies. Baker is planning to steal cookies from the boxes to share with his friends! To share with all his friends, he needs N chocolate chip cookies. After exploring the pantry, Baker noticed that the boxes come in 4 types:

- Black boxes, which contain only chocolate chip cookies.
- White boxes, which contain only oatmeal cookies.
- Red boxes, which contain some chocolate chip cookies.
- Green boxes, which contain some oatmeal cookies

Baker plans to steal the cookies one at a time. On each trip to the pantry he will select a box and reach into that box to take a cookie – the cookie will be chosen at random from the box's contents, but Baker will always leave the pantry with exactly one cookie. He will repeat this process as many times as it takes to acquire N chocolate chip cookies.

Before carrying out his plan, Baker needs your help to determine the minimum number of trips M he will need to make to the pantry to guarantee that he will acquire at least N chocolate chip cookies, assuming he selects the boxes optimally. If it is impossible to guarantee that he can steal N chocolate chip cookies, instead advise him "Don't even try!"

Input

The first line of input contains a single integer T, representing the number of test cases. Each of the following T lines describes a test case with a single line containing 6 integer numbers separated by spaces, N, C, B, W, R, G, representing, respectively, the desired number of chocolate cookies, the number of cookies in each box, the number of black boxes, the number of white boxes, the number of red boxes, and the number of green boxes in the pantry.

- $1 \leq T \leq 100$
- $1 \leq N \leq 50000$
- $1 \leq C, B, W, R, G \leq 100$

Output

For each test case print the minimum number of trips (M) Baker will need to make to the pantry to guarantee that he will get at least N chocolate chip cookies. Or, if it is impossible to guarantee that Baker will get at least N chocolate chip cookies, print "Don't even try!" (without quotes).

Input example 1	**Output example 1**

`3`
`5 10 1 0 0 0`
`2 10 0 2 1 0`
`52 7 7 7 7 7`
	`5`
	`Don't even try!`
	`70`

Problem J
Elisa's Playroom

Elisa has N toys, numbered from 1 to N, that she uses to play while her parents are working, her parents have noticed that there are some pairs of toys that she likes to combine while playing, and, she gets sad if for any reason she can find one of those toys in her playroom but not the other. This is, if Elisa likes to combine toys 1 and 2 while playing, and toy 1 is in the playroom but not toy 2 she will get sad. Also, suppose Elisa likes to combine toys 1 and 2, and also toys 2 and 3, if one of those toys is in the playroom and any of the other two is not, she will get sad.

Elisa's parents do not like to see her toddler sad, that is why they decided to put M toys in the playroom making sure that Elisa will be able to find any toy she looks for when playing. To do this, Elisa's parents have a complete list of the pairs of toys Elisa likes to combine when playing. Help Elisa parents find all possible distinct values for M they could use when setting the playroom for Elisa.

Input

The first line of input contains two integer numbers separated by a space, N, and P, representing, respectively, the number N of toys Elisa has, and the number of P pairs she likes to combine when playing. Each of the following P lines contains two integer numbers u, and v, representing Elisa likes to combine toys u and v when playing. You can assume if Elisa likes to combine u and v, she likes to combine v and u.

- $1 \leq N \leq 1000$
- $0 \leq P \leq \frac{(N)(N-1)}{2}$
- $1 \leq u, v \leq N$, and $u \neq v$
- It is guaranteed no two pairs are repeated in the list.

Output

Print a line with a single integer, the amount of distinct values for the number M of toys, Elisa's parents can put in the playroom ensuring Elisa will not get sad when playing.

Input example 1	Output example 1

`4 2`
`1 2`
`1 3`
	`3`

Input example 2	Output example 2
`4 0`	`4`

Input example 3	Output example 3

`3 2`
`1 2`
`2 3`
	`1`

Input example 4	**Output example 4**

`4 2`
`1 2`
`3 4`
	`2`

Notes

In the first test case there are three different values for $M = 1, 3, 4$. There is no way in which putting 2 toys in the playroom Elisa does not get sad.

Problem K
Lumberjacks

Another tale of lumberjacks?. Let see ...

The lumberjacks are rude, bearded workers, while foremen tend to be bossy and simple-minded. The foremen like to harass the lumberjacks by making them line up in groups of ten, ordered by the length of their beards. The lumberjacks, being of different physical heights, vary their arrangements to confuse the foremen. Therefore, the foremen must actually measure the beards in centimeters to see if everyone is lined up in order.

Your task is to write a program to assist the foremen in determining whether or not the lumberjacks are lined up properly, either from shortest to longest beard or from longest to shortest.

Input

The first line of input contains a single integer T, representing the number of test cases. Each of the next are T lines describes a test case, each test case contains 10 distinct positive, representing the measurement of the beards of the lumberjacks in the line.

- $1 \leq T \leq 20$.
- The beard of each lumberjack is an integer number between 0 and 100.

Output

For each test case print a line with the word "Ordered" if the lumberjacks in the line are liner properly, otherwise, print the word "Unordered".

Input example 1	**Output example 1**

`3`
`13 25 39 40 55 62 68 77 88 95`
`88 62 77 20 40 10 99 56 45 36`
`91 78 61 59 54 49 43 33 26 18`
	`Ordered`
	`Unordered`
	`Ordered`

Problem L
Last Younger Person

Planet E-13 orbits around a star in a faraway galaxy named UAZ. On December 18th, there will be a massive concert by one of the most famous rock bands in the universe, the venue where the concert will take place has a capacity of 500000 persons and there will be a single line to enter the concert. As people enter the concert, there is a guard named Peter, asking each person that enters the venue, his/her name and birth date, which Peter must keep somehow (along with the place number each person had in the line). Peter needs this information, because Thanitos, the concert organizer, for some strange reason (maybe he just wants to make Peter suffer), would like to know, for a person in place q in the line, the name C_r and place number r in the line of the last younger person to enter before person q. A person r is considered younger than person q, if the birthdate of person r is at least one day before than that of person q. Thanitos could ask such questions for only one of the persons that entered to the venue, or for all of them, and in any order.

Your work is to help Peter to answer such queries from Thanitos as fast as possible.

Input

The first line contains an integer number N representing the number of persons that will attend the concert. The following N lines have the name C_i and birth date B_i for each person i attending the concert, separated by a space, according to its place on the line (the front of the line is place number 1). The next line contains an integer number M representing the number of queries Thanitos will make to the guard. The last line will contain M numbers q_i representing the place numbers in which Thanitos is interested.

- $1 \leq N \leq 500000$
- $1 \leq M \leq N$
- $1 \leq q_i \leq N$
- The name C_i is comprised of only lowercase latin letters and with a maximum length of 10.
- The birtdate of each persion B_i is in the format YYYY/MM/DD. Where YYYY represents the year with possible values from 0000 to 9999, MM represents the month number with possible values from 01 to 20, and DD represents the day in the month, with possible values from 01 to 50.

Output

The output consists of M lines indicating for each of the queries q_i, either if no one younger arrived before the person at place q_i (including its name C_{q_i}) or if someone younger arrived before, the name of the last younger person that arrived before the person at place q_i (including the place number that such younger person had in the line), with the following format: At place $\#q_i$, no one younger arrived before C_{q_i} or At place $\#q_i$, the last younger person to arrive before C_{q_i} is $C_r(\#r)$. Where r is the position of the last younger person who arrived before C_{q_i}

Input example 1

```
8
roberto 1971/01/15
juan 1980/02/17
maria 1965/12/23
roberto 1977/17/39
julieta 1991/13/46
juan 1976/20/05
ramiro 1971/01/16
oscar 1972/10/46
6
1 7 3 2 5 4
```

Output example 1

```
At place #1, no one younger arrived before roberto
At place #7, the last younger person to arrive before ramiro is maria(#3)
At place #3, no one younger arrived before maria
At place #2, the last younger person to arrive before juan is roberto(#1)
At place #5, the last younger person to arrive before julieta is roberto(#4)
At place #4, the last younger person to arrive before roberto is maria(#3)
```