2973 - Digit Sum

Description

When Grace was in third grade, her elementary school teacher assigned her the following problem:

- What is the smallest possible sum of two numbers that together use the numerals 1, 2, 7, 8, and 9?

Grace figured out that the answer to this problem is 207 (for example, as 78 + 129), but when the teacher assigned four pages of similar problems as homework, Grace got bored. It turns out that Grace was a rather advanced third grader, so she decided that it would be more fun to write a computer program to solve such problems. Surely you can do the same!

Input specification

Each problem is described on a single line. The line begins with an integer N, such that 2 <= N <= 14, designating the number of numerals included in the problem. Following that are those N numerals. There will always be at least 2 numerals that are nonzero. The end of the input is designated by a line containing only the value 0.

Output specification

For each case, output a line with the minimum sum S that can be achieved. Please keep in mind that by standard convention, the numeral 0 cannot appear as the first digit of either summand.

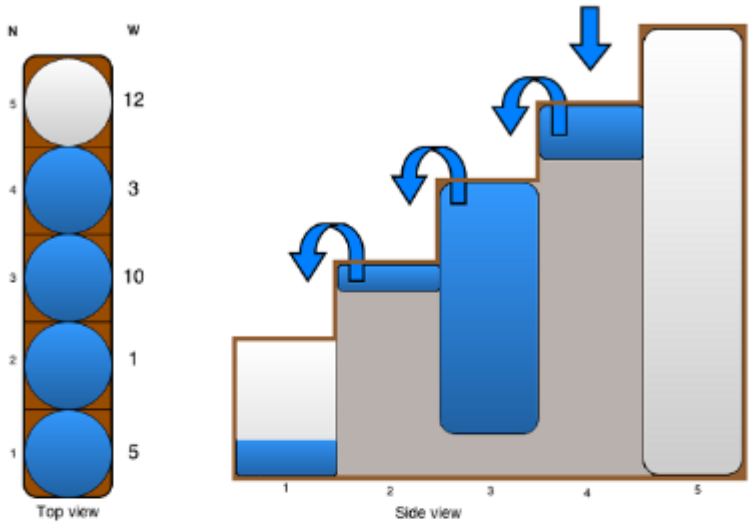
Sample input

```
5 1 2 7 8 9
6 3 4 2 2 2 2
9 0 1 2 3 4 0 1 2 3
0
```

Sample output

```
207
447
11257
```

Tobby is a curious dog and he is always thinking about new and revolutionary ideas. Now he imagined N tanks placed in a row (staggered - See the figure). Every tank has a capacity of W liters of water. The cute and curious dog is asked himself, if you have K liters of water, which is the i-th tank (where i is the maximum possible), such that if we drop the water there, the water will reach the first tank. As you know Tobby is not a complicated dog, so he does not want completely fill the first tank, he will be happy if the first tank have at least 1 liter of water.



You are given n tanks with capacity w_i, and q queries, each query contains one integer k, the amount of water.

Input specification

For every test case, the first line contains two integers n (1 <= n <= 10^5) and q (1 <= q <= 10^4), the number of tanks and the number of queries respectively. The next line contains n integers w_i (1 <= w_i <= 10^4). The next line contains q integers numbers k_j (1 <= k_j <= 10^9), the amount of water. You must read until you reach the EOF.

Output specification

You must print q space-separated integers numbers in a single line, one per query, with the maximum possible value.

Sample input

```
5 2
5 1 10 3 12
16 1
```

Sample output

```
4 1
```

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-06-12
Total Time	4000
Memory	256000
Output	64
Size	15000

2930 - Assembly I

Description

This year some Colombian university have decided to remodel the access to the university, and for giving them a modern touch, the Superior Council of this university chose to put some control access devices for avoiding strangers and thiefs to make harm to students and facilities.

In this university there are a lot of left wing political groups, and the main feature of this kind of groups is to be against almost any decision taken by the Superior Council, and the decision about control access devices will not be the exception, obviously. For deciding wich actions will be taken against the Superior Council decisions, these political students group gather all the other students in an event called student assembly, which you can modelate in the following way:

- There is an speech sequence of size N and N speaker students which are the ones who persuade the other students to take position against or in favor of an initiative from Superior Council. Each speech in the sequence corresponds to only one speaker student.
- For each speech i there is an acceptance level S (0 <= S <= 100) after it. If S = 0, then, all the students are against the superior council decision, but if S = 100, then, all the students will accept the superior council decision.

The university Headmaster has found out about students intentions. For that reason, he decided to sneak in the students assembly disguised as a revolutionary student, and not only that, but also he carried with him a sensor designed by the enigneering school which allows him to know in real time each speech acceptance level.

You as a clever student of the computer science program find out the mean intentions of the headmaster, so, after some minutes of hacking the new headmaster's toy, you realize that the sensor works in the following way:

- The device will show to the headmaster a message saying "Caution", if the longest increasing subsequence of speeches is equally larger than the longest decreasing subsequence of speeches.
- The device will show to the headmaster a message saying "Don't worry", if the longest increasing subsequence of speeches is strictly less or strictly greater than the longest decreasing subsequence.
- An increasing subsequence is defined as a subset(not necesarily continuous) S^* = [a_1, a_2, ..., a_k] from the original sequence of speeches S = [S_0, S_1, ..., S_N], such that if i > j, then, a_i > a_j. Here a_i is acceptance level for an speech within the subsequence, and S_i is the acceptance level in the original sequence.
- A decreasing subsequence is defined as a subset(not necesarily continuous) S^* = [b_1, b_2, ..., b_k] from the original sequence of speeches S = [S_0, S_1, ..., S_N], such that if i > j, then b_i < b_j. Here b_i is the acceptance level in the subsequence.

For example, supose you have the next sequence of speeches [10, 25, 20, 22, 90, 21], one increasing subsequence could be [10, 90] and one decreasing subsequence could be [25, 20] but this both are not necesearily the longest ones. In this case the longest increasing subsequence is [10, 20, 22, 90] and the longest decreasing subsequence is [25, 22, 21], look that in this case the output message from the sensor will be "Don't worry".

After knowing this, you realize that the sensor used by the headmaster have an algorithm for finding this subsequences that runs in O(N!) time, and while you are thinking why some profesors of the engineering school that designed de sensor are so lazy, you also wonder if you can design a better algorithm to know before the headmaster what is the outcome of the sensor.

Given that outcome you will decide if you must or must not intervene in the assembly, either for avoiding a strike or avoiding people to forget the university issues.

Input specification

The input will have several test cases, in the first line of each test case there will be an integer N (1 <= N <= 1000) corresponding to the amount of speeches. The second line will have N integer numbers S_1, S_2, ..., S_i, S_N, where S_i (0 <= s_i <= 100) represents the acceptance level for the speech i.

Output specification

There are only two options of output

- If the longest increasing subsequence is equal in length as the longest decreasing subsequence you must print "Caution. I will not intervene." representing the outcome of the sensor and what you will probably do.
- If the longest increasing subsequence is NOT equal in length as the longest decreasing subsequence you must print "Don't worry. I must intervene."

Sample input

```
6
10 25 20 22 90 91
5
10 10 10 10 10
5
1 20 50 49 48
```

Sample output

```
Don't worry. I must intervene.
Caution. I will not intervene.
Caution. I will not intervene.
```

Hint(s)

Given a number in the original sequence you can know easily whether the numbers before the given number are less or greater than it, and store that info in someway. take care about the number of operations, and take into account that sometimes there are data already calculated that maybe you do not need to calculate again.

2907 - Lineland

Description

Lineland has always been caracterized as a prolific civilization but in this moment, thanks to what seems to be an attack of an individual, The electrical network from lineland has been damaged, they don't know where the responsible came from, maybe from flatalandd, from cubicland or even tobyland. Well, the fact is that now the city managers from lineland only can use one electrical plant from from the N possible electrical plants installed in lineland. So they have to choose an electrical plant which is going to be able to give energy to maximum amount of cities. can you help the managers from lineland with this problem?.

Lineland can be modelate as a set of points on an horizontal axis, each point will represent an electrical plant ubicated in one city. Besides, one electrical plant ubicated in a point x_i with a capacity c_i, will be able to supply a city ubicated in x_j if, and only if, x_i < x_j < x_i + c_i. Your task is to choose the electrical plant which can suply the maximum number of cities.

Input specification

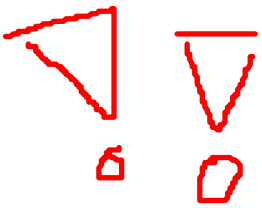
The input will have several test cases, in the first line of each test case there will be an integer N (1 <= N <= 10^6), In the next N lines there will be two integers x and c (0 <= x <= 10^9, 0 <= c <= 10^9); x_i represents the position of the i-th city and the i-th plant, while c_i represents the capacity of the i-th plant (1 <= i <= N). You might assume this two tips:

- No two cities i and j are ubicated in the same location.
- For each two cities ubicated in x_i, x_j (i != j) with capacity c_i and c_j that can supply another cities within a range of (x_i, x_i + c_i) and (x_j, x_j + c_j) respectively, this two conditions hold: x_i + c_i != x_j + c_j and x_i + c_i != x_j.

Output specification

You have to print two integers in one line, that is, the position of the electric plant that will supply the maximum amount of cities, followed by an space and the respective amount of cities that will be supplied by that electric plant.

Sample input



Sample input

```
7
1 7
5 4
6 5
7 15
10 3
12 5
16 3
5
1 1
3 2
6 3
10 4
15 22
3
1000 1000000000
1003 10000
50000 1000000000
```

Sample output

```
1 3
1 0
1000 2
```

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-05-10
Total Time	5000
Memory	256000
Output	64
Size	15000

2974 - Sort Me

Description

We know the normal alphabetical order of the English alphabet, and we can then sort words or other letter sequences. For instance these words are sorted:
ANTLER
ANY
COW
HILL
HOW
HOWEVER
WHATEVER
ZONE

The standard rules for sorting letter sequences are used:

1. The first letters are in alphabetical order. Among strings with the same prefix, like the prefix AN in ANTLER and ANY, they are ordered by the first character that is different, T or Y here.
2. One whole string may be a prefix of another string, like HOW and HOWEVER. In this case the longer sequence comes after the shorter one.
3. The Gorellians, at the far end of our galaxy, have discovered various samples of English text from our electronic transmissions, but they did not find the order of our alphabet. Being a very organized and orderly species, they want to have a way of ordering words, even in the strange symbols of English. Hence they must determine their own order. Unfortunately they cannot agree, and every Gorellian year, they argue and settle on a new order.

For instance, if they agree on the alphabetical order UVWXYZNOPQRSTHIJKLMABCDEFG then the words above would be sorted as
WHATEVER
ZONE
HOW
HOWEVER
HILL
ANY
ANTLER
COW

The first letters of the words are in their alphabetical order. Where words have the same prefix, the first differing letter determines the order, so the order goes ANY, then ANTLER, since Y is before T in their choice of alphabet. Still HOWEVER comes after HOW, since HOW is a prefix of HOWEVER. Dealing with the different alphabetical orders each year by hand (or tentacle) is tedious. Your job is to implement sorting with the English letters in a specified sequence.

Input specification

The input will contain one or more datasets. Each dataset will start with a line containing an integer n and a string s, where s is a permutation of the English uppercase alphabet, used as the Gorellians' alphabet in the coming year. The next n lines (1 <= n <= 20) will each contain one non-empty string of letters. The length of each string will be no more than 30. Following the last dataset is a line containing only 0.

Output specification

The first line of output of each dataset will contain "year " followed by the number of the dataset, starting from 1. The remaining n lines are the n input strings sorted assuming the alphabet has the order in s.

Sample input

```
8 UVWXYZNOPQRSTHIJKLMABCDEFG
ANTLER
ANY
COW
HILL
HOW
HOWEVER
WHATEVER
ZONE
5 ZYXWVUTSRQPONMLKJIHGFEDCBA
GO
ALL
ACM
TEAMS
GO
10 ZOTFISENWABCDGHJKLMPQRUVXY
THREE
ONE
NINE
FIVE
SEVEN
ZERO
TWO
FOUR
EIGHT
SIX
0
```

Sample output

```
year 1
WHATEVER
ZONE
HOW
HOWEVER
HILL
ANY
ANTLER
COW
year 2
TEAMS
GO
GO
ALL
ACM
year 3
ZERO
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE
```

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-07-10
Total Time	1000
Memory	256000
Output	64
Size	15000

2968 - Tobby and the River

Description

Tobby is in one side of a river with 5 other dogs. In that side there are two small boats in which Tobby and the other five dogs can cross to the other side. Tobby knows that in order to cross to the other side the weight in both boats must be as balanced as possible, because if it is not the boats might sink.

Tobby wants to know if there is a way to divide the 6 dogs (he and the other 5 dogs) in both boats in such a way that both boats have the same weight. Both boats don't need to have the same amount of dogs, but all the 6 dogs must be in one of the two boats.

Input specification

In the first line there is a number A (1 <= A <= 100), after that there are A lines with 6 positive integer numbers less than or equal to 100.

Output specification

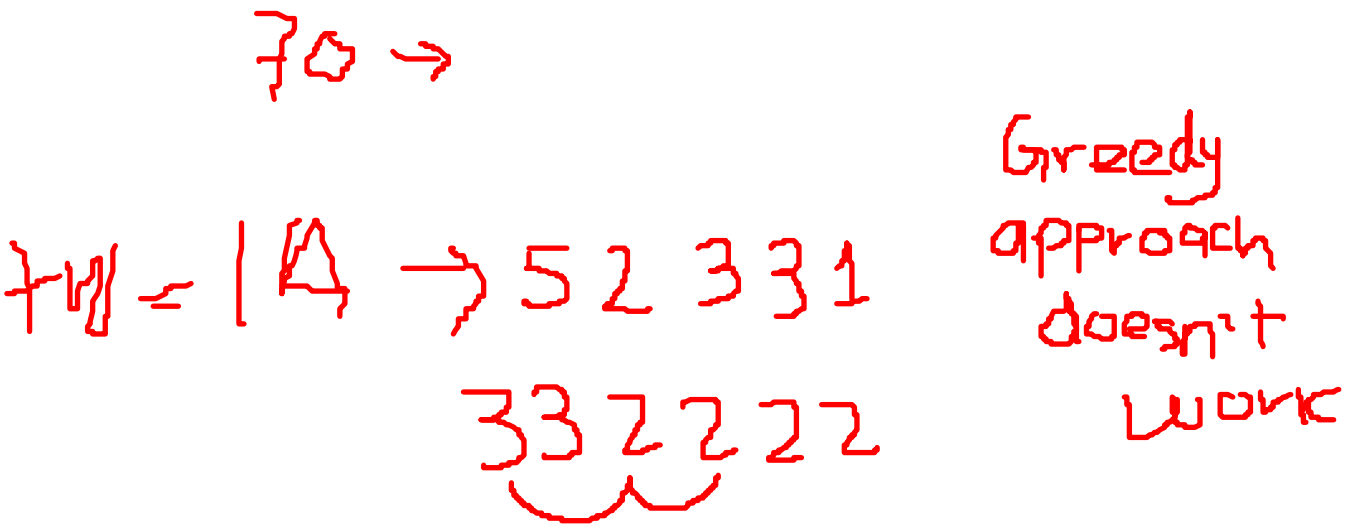
For each one of the input lines you must write a line with the spanish sentence "Tobby puede cruzar" if there is a way of dividing the 6 dogs in both boats with the same weight or the spanish sentence "Tobby no puede cruzar" otherwise.

Sample input

```
4
1 3 3 2 1 2
6 3 2 4 5 1
3 3 3 3 3 3
1 1 1 1 1 5
```

Sample output

```
Tobby puede cruzar
Tobby no puede cruzar
Tobby puede cruzar
Tobby puede cruzar
```

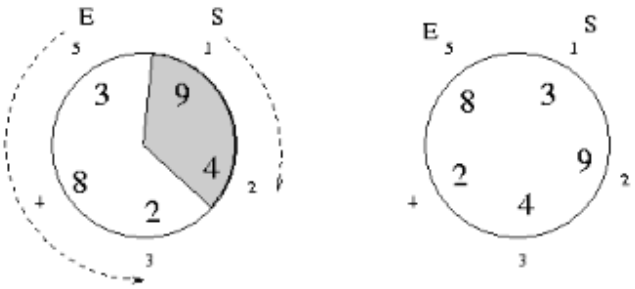


2976 - Magnificent Meatballs

Description

Sam and Ella run a catering service. They like to put on a show when serving meatballs to guests seated at round tables. They march out of the kitchen with pots of meatballs and start serving adjacent guests. Ella goes counterclockwise and Sam goes clockwise, until they both plop down their last meatball, at the same time, again at adjacent guests. This impressive routine can only be accomplished if they can divide the table into two sections, each having the same number of meatballs. You are to write a program to assist them.

At these catering events, each table seats 2 <= N <= 30 guests. Each guest orders at least one and at most nine meatballs. Each place at the table is numbered from 1 to N, with the host at position 1 and the host's spouse at position N. Sam always serves the host first then proceeds to serve guests in increasing order. Ella serves the spouse first, then serves guests in decreasing order. The figures illustrate the first two example input cases.



Input specification

Input consists of one or more test cases. Each test case contains the number of guests N followed by meatballs ordered by each guest, from guest 1 to guest N. The end of the input is a line with a single zero.

Output specification

For each table, output a single line with the ending positions for Sam and Ella, or the sentence indicating an equal partitioning isn't possible. Use the exact formatting shown below.

Sample input

```
5 9 4 2 8 3
5 3 9 4 2 8
6 1 2 1 2 1 2
6 1 2 1 2 1 1
0
```

Sample output

```
Sam stops at position 2 and Ella stops at position 3.
No equal partitioning.
No equal partitioning.
Sam stops at position 3 and Ella stops at position 4.
```

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-07-10
Total Time	1000
Memory	256000
Output	64

2932 - Tobby and the Primoshkas' Tree I

Description

Tobby just found a very special tree. This kind of tree produces a magical creature called Primoshka. Primoshkas are little and quite funny creatures that entertain people in several ways. The operation of this tree is a little weird, First of all, if someone wants to collect Primoshkas he/she must select a fork in the tree and tell it one of the following phrases:

- "Please create new primoshkas", in this case for each fork below, the selected fork (the selected fork is affected too), a new primoshka arises. In case that one primoshka already exists in a determined fork, she falls and dissapears. You could assume both of them will fall. In other words, if a fork has one primoshka, it will not have primoshkas after this operation. It's like a toggle operation.
- "Please collect primoshkas", in this case all primoshkas below to the current fork (and also in the selected fork) are collected, however, due to unknown reasons a bunch of new primoshkas appear after that, but only in the forks where primoshkas previously existed (Leaving the tree as if nothing had changed).

Initially the tree is empty. The tree just have one root, in others words, there are exactly one fork which have no forks above it. Furthermore the tree don't have any kind of cycles.

Input specification

The input consists of several test cases. The test case begins with two numbers 3 <= N <= 100 and 1 <= Q <= 100 followed by N - 1 lines with a pair of numbers u_i, v_i which indicate that the fork number u_i and the fork number v_i are connected (v_i is below u_i).

After that, Q lines with a pair of integers op_i and f_i which indicates the type of the operation (0 for the first phrase and 1 for the second phrase) and the number of the selected fork. All forks are labeled between 0 and N - 1.

Output specification

For each one of the second type operations you must print the number of collected primoshkas in that operation.

Sample input

```
5 4
0 1
1 2
1 3
0 4
0 1
1 1
0 0
1 0
```

Sample output

```
3
2
```

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-05-26
Total Time	1000
Memory	256000
Output	64
Size	15000

2970 - Humbertov's Flag

Description

Humbertov Moralov wants to create a new flag for his empire, probably because he is thinking to invade Tobyland again. He doesn't want a common rectangular flag and he has ordered you to find new designs. You have to show Moralov a good amount of different designs quicIky or Moralov is going to ask for your head.

The new designs should be polygons, not only convex polygons, but you certainly don't want self intersecting polygons because you can't translate them to real flag designs. So you decide to write a program to generate non self-intersecting polygons, so you decided to write a program that generates these polygons for you. Given a set of points, your program must output a non self-intersecting polygon that includes every one of the given points.

Input specification

The input consists to several test cases. Each test case begins with a integer 3 <= N <= 100 indicating the number of points in the list. Then come N lines each with two integers indicating the x and y coordinates of each point of the list (-100000 <= x_i, y_i <= 100000).

Output specification

For each test case, print N followed by N lines each with one of the points read on the input, but you can change the order in any way you like such that the resulting polygon is not self intersecting.

Sample input

```
4
1 1
-1 -1
1 -1
-1 1
3
0 0
0 2
2 0
```

Sample output

```
4
1 1
-1 1
-1 -1
1 -1
3
0 0
0 2
2 0
```

Concave hull

Hint(s)

Created by	TBA
Added by	TBA
Addition date	2014-07-10
Total Time	1000
Memory	256000
Output	64
Size	15000
Enabled languages	Bash C C# C++ Java Pascal Perl PHP Python Ruby