CODING NOW PROGRAMMING FUTURE

4th Jilin Province Collegiate Programming Contest

第四届吉林省大学生程序设计竞赛

May 22, 2010

1	Problem A	Harshad Number
2	Problem B	The Farmer's Heritage
3	Problem C	The Natural Series
4	Problem D	Comparing Answers
5	Problem E	Warehouse and Truck
6	Problem F	Coin Game
7	Problem G	Planar Graph
8	Problem H	New Feature in Windows 7
9	Problem I	Duodecimal
10	Problem J	Buying and Selling

Notice: If the content has been patched, please refer to the web page
注：如有修改，以网上实时页面为准。

Problem A: Harshad Number

Welcome to 2010 Jilin Province Collegiate Programming Contest. Do you know 2010 is a magic number? If you add up all the digits of 2010, that is 3, it is just one factor of 2010. The number who has this character is called Harshad number.

A Harshad number, or Niven number in a given number base, is an integer that is divisible by the sum of its digits when written in that base. Harshad numbers were defined by D. R. Kaprekar, a mathematician from India. The word "Harshad" comes from the Sanskrit "harṣa", meaning "great joy". The Niven numbers take their name from Ivan M. Niven from a paper delivered at a conference on number theory in 1997. All integers between zero and n are Harshad numbers in base n.

The Harshad number is rare in all natural numbers. The ratio is about 10% decreasing with the number value.

Your task is to list all the Harshad number between 1000 and 9999 orderly. We just consider the normal decimal base.

Input

This problem has no input

Output

You should output each Harshad number in one line orderly.

Sample Output

```
1000
......
2010
......
9990
```

Not all the numbers are listed in the sample. There are just the first, the last, and 2010 itself. The ellipsis expresses what you should calculate.

Problem B: The Farmer's Heritage

A farmer who had worked hard all his life on his triangular farm was taken sick. He knew that he must soon die. He called his three sons about his bed to give them some advice. "My sons," said he, "keep all of the land which I leave you. Do not sell any of it, for there is a treasure in the soil. I shall not tell you where to hunt for it, but if you try hard to find it, and do not give up, you will surely succeed. As soon as the harvest is over, begin your search with plow, and spade, and rake. Turn every foot of earth, and then turn it again and again. The treasure is there."

After the father died, the sons gathered in the harvest. As soon as the grain had been cared for, they planned to search for the hidden treasure. The farm was divided into three small triangles. Each son agreed to dig carefully his part. Every foot of soil was turned by the plow or by the spade. It was next harrowed and raked, but no treasure was found. That seemed very strange.

"Father was an honest man and a wise man," said the youngest son. "He would never have told us to hunt for the treasure if it were not here. Do you not remember that he said, 'Turn the soil again and again'? He surely thought the treasure worth hunting for."

"Our land is in such good condition now that we might as well sow winter wheat," said the oldest son. His brothers agreed to this and the wheat was sown.

The next harvest was so great that it surprised them. No neighbor's field bore so many bushels of wheat to the acre. The sons were pleased with their success.

This story tells us that we must work hard to success. And do you know how these three sons divide the farm? Because of their ages, three sons acquired different are of this heritage. The eldest brother should get 1/2, the second should get 1/3, and the youngest son get 1/6.

Input

There are x and y coordinates of three triangle vertices in one line. These six numbers are all real number. The edge against the first vertex belongs to the eldest brother, the edge against the third vertex belongs to the youngest, and so the second. For each case, you should find a point inside the triangle that can divide the original triangle to three equal triangles.

Output

Output the x and y coordinates of the expected point in one line separate by a space. Keep 3 digits after the decimal point.

Sample Input

```
0.0 0.0 0.0 6.0 6.0 0.0
```

Sample Output

```
1.000 2.000
```

Problem C: The Natural Series

The natural series, from 1 to n and rewinding, has a special character. If you add two values whose distance is 2, the sum can be divided by the middle value exactly. Let's check a example "1 2 3 4 5 6" (and rewinding to 1): (1+3)%2==0; (2+4)%3==0; … ;(5+1)%6==0, and (6+2)%1==0. This character is unusual. You can't find another series when n is even, and only one different series essentially when n is odd. For example, when n is 7, the other one is "1 2 3 7 4 5 6".

Here we relax the constraints. If the sum of two numbers is less than the middle value, this series can be accepted also. For example, "1 2 7 4 9 5 11 3 10 6 8" is true where 5+3<11.

Input

Each line of input has two numbers, the first is *n (4<=n<=16)*. The next positive integer *m (1<m<=n)* is a reference that you must use it to be the second number in result series.

Output

Print each series under above conditions in one line in sequence. The first number must be 1, and second is the given reference. There is a blank between adjacent numbers in one line and no blank after the last number. Print a blank between each case and also no blank line after last case.

Sample Input

```
10 4
7 2
```

Sample Output

```
1 4 2 6 10 3 8 5 7 9
1 4 2 8 6 10 3 5 7 9

1 2 3 4 5 6 7
1 2 3 7 4 5 6
```

Problem D: Comparing Answers

Multiple-choice is a form of assessment in which testers are asked to select the best answer out of the choices from a list. The multiple choice format is most frequently used in educational testing.

Last week, Skywind has attended an English test that is full of multiple choice questions. After the examination, Skywind could remember how many "A, B, C, D" that he chose. Furthermore, he could remember a set of possible selections for each question although only one answer is right. For example, he knew that he might select 'A' or 'C' for question 1 but not 'B' and 'D'. After the examination, Skywind got the standard answer. In most case, he can't calculate the exact score that he could gain because of his poor memory. What he want know urgently is the highest score and the lowest score that he could gain possible.

These are n multiple choice questions. Two mark will be given if the answer is right, none if wrong.

Input

The first line contains the number of test cases you will process.

In each test case, the first line is *n(n<=300)* representing the number of questions, followed by four integers which are how many "A, B, C, D" Skywind chose. Then following string whose length is *n* represents the standard answer. The *i-th* line expresses his memory of *i-th* question in the next n lines. In each line of this section, there are one number *k* and *k* characters representing the possible answer for this question.

Output

Print the highest score and lowest score he may get in one line divided by a space.

If a conflict is found between the memory of each question and how many "A, B, C, D" he chose totally, you should output "Forgetful".

Sample Input

```
3
4
1 1 1 1
ABCD
1 A
```

```
1 B
2 C D
2 C D

4
4 0 0 0
ACCB
2 A B
1 A
3 A B D
2 A D

4
4 0 0 0
ACBD
1 B
1 A
1 A
1 A
```

Sample Output

```
8 4
2 2
Forgetful
```

Problem E: Warehouse and Truck

There is a warehouse located at the origin of a straight road and some stores scattered on this road. Every day, one truck is laden with all of the goods needed by these stores from the warehouse and then delivers goods to every store in turn. The gas consumed by this truck varies directly with the product of goods weight and driving distance. That is, if the weight of goods is x now and the truck runs y miles, the gas burned is x*y. Here, the factor of truck itself is omitted.

As you wish, you are asked to find the best strategy to save gas as much as possible. The known conditions are the places and requirements of stores.

In addition, the weight that the truck loads initially is just the sum of requirements. That is to say, the truck can't reload even if it drives across the warehouse again.

Input

The first line of each case is the number of stores *n (1<=n<=100)*. There are two integers identified the coordinate and requirement in the following *n* lines. Of course, the requirement is positive, and the coordinate can be positive or negative. The absolute values of these parameters are less than *10000*.

Output

For each case, you should output gas used under the best strategy in each line.

Sample Input

```
4
1 1000
-1 900
100 1
-100 1
```

Sample Output

```
4104
```

Problem F: Coin Game

Gordon and Lee, fond of collecting coins, are playing a coin game.

At first, Gordon puts N piles of his coins, decreasing-order in heights, in a row. Then Lee also puts N piles of his coins, randomly, in another row. The game rules are as follows: Every time Lee can swap any two adjacent piles of his coins. Lee, if successfully making each pile of his coins NOT higher than the correspondent pile of Gordon's in the given time, can win all the coins away. Lee, a smart coder, is planning to write a program to calculate the minimum swaps he needs to win the game.

Input

Each test case starts with a line containing an integers *N (1<=N<=100)* which is the number of piles. The next two lines each contain N integers, namely the heights of each pile of coins of Gordon's and Lee's, respectively. You can assume there is always a feasible solution for Lee to win the game.

Output

Print the minimum swaps Lee needs to win the game.

Sample Input

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

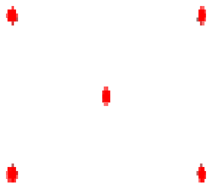
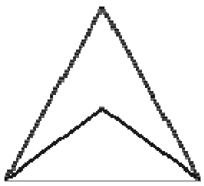
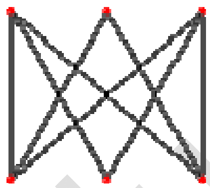
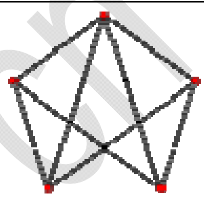
Sample Output

```
0
3
5
```

Problem G: Planar Graph

In graph theory, a planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. A planar graph already drawn in the plane without edge intersections is called a plane graph or planar embedding of the graph.

The following figure show some planar and nonplanar graphs.

Planar Graphs		Nonplanar Graphs	
Butterfly graph	Complete Graph 4	K3,3	K5
			

The Polish mathematician Kazimierz Kuratowski provided a characterization of planar graphs in terms of forbidden graphs, now known as Kuratowski's theorem:

A finite graph is planar if and only if it does not contain a subgraph that is a subdivision of K5 (the complete graph on five vertices) or K3,3 (complete bipartite graph on six vertices, three of which connect to each of the other three).

A subdivision is a special operation that we don’t consider here. What we should do is to determine whether a K5 or K3,3 can be found in a graph.

Clearly, given a graph, can you find 5 vertices which connect each other? Can you find six vertices, three of which connect to each of the other three?

Input

The first line of each case is two integers m $(2<m<=100)$, n $(1<=n<=m*(m-1)/2)$. Here, m is the number of vertices in the graph, n is the number of edges. The following n lines contains two vertices index (start from 0) identified this edges.

Output

For each test case, output two characters divided by a space in one line. If K5 is existed, the first character is ‘Y’, otherwise is ‘N’; if K3,3 is existed, the second character is ‘Y’, otherwise is ‘N’.

Sample Input

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

Sample Output

```
Y N
```

Problem H: New Feature in Windows 7

Windows 7 is one version of Microsoft Windows Family; a series of operating systems produced by Microsoft running on personal computers, including home and business desktops, laptops, notebooks, tablet PCs, and media center PCs. Windows 7 was released to manufacturing on July 22, 2009, and reached general retail availability on October 22, 2009.

Windows 7 studies and includes some features from some other successful operating systems, such as Mac OS and Linux. It is said that Windows 7 is more user-friendly than its predecessors especially than Vista, using which you may be forced to click “OK” four times just for copying files.

For example, you can find something more interesting when sorting files. Of course, files will be sorted by their names in alphabetical order in some version of previous Windows. However, it may bring unexpected results. Suppose that there are some files named as “Lesson1”, “Lesson2”, …, “Lesson9”, ”Lesson10” and “Lesson11”, in which the numbers describe the sequence of lessons. Then, in previous version the numbers in filenames are ranked by alphabetical place and the sorted results are “Lesson1”, ”Lesson10”, “Lesson11”, “Lesson2”, …“Lesson9”. Obviously, it is not what you want, because the numbers should be ranked by their values. Now, Windows 7 realizes this.

Now your task is to sort these filenames from small to large like Windows7. Exactly speaking, if two filenames are same before certain position including start, the order should be decided by these conditions:

a) If both of them are alphabets, the result depends on the alphabet order.

b) If one is alphabet and the other is number, the number is smaller.

c) If both of them are numbers, the result depends on number values.

To simplify the problem, firstly, only filenames excluding extensions are given. Secondly, only alphabets ‘A’-‘Z’, ‘a’-‘z’, digits ‘0’-‘9’ are used in filenames. Finally, there is no such number whose first digit is ‘0’.

Input

The first line of each case is an integer n $(2 \le n \le 100)$, which is the number of the following filenames. The end of input is indicated by n=0.

There is only one filename in each following lines. The length of filename is no more than 100 characters.

Output

For each case, sort these filenames according to the above rules. Print the results on every line. Output a blank line after each case.

Sample Input

```
3
Lesson11
Lesson2
Lesson1
5
A101b
A1
B1
A2a
A99b
0
```

Sample Output

```
Lesson1
Lesson2
Lesson11

A1
A2a
A99b
A101b
B1

```

Problem I: Duodecimal

The duodecimal system (also known as base-12 or dozenal) is a positional notation numeral system using twelve as its base. In this system, the number ten may be written as 'A' or 'X', and the number eleven as 'B' or 'E'. The number twelve (that is, the number written as '12' in the base ten numerical system) is instead written as '10' in duodecimal, whereas the digit string '12' means "1 dozen and 2 units" (i.e. the same number that in decimal is written as '14').

The number twelve, a highly composite number, is the smallest number with four non-trivial factors (2, 3, 4, 6), and the smallest to include as factors all four numbers (1 to 4). As a result of this increased factorability of the radix and its divisibility by a wide range of the most elemental numbers (whereas ten has only two non-trivial factors: 2 and 5, with neither 3 nor 4), duodecimal representations fit more easily than decimal ones into many common patterns, as evidenced by the higher regularity observable in the duodecimal multiplication table. Of its factors, 2 and 3 are prime, which means the reciprocals of all 3-smooth numbers (such as 2, 3, 4, 6, 8, 9...) have a terminating representation in duodecimal.

In particular, the five most elementary fractions ($1/2$, $1/3$, $2/3$, $1/4$ and $3/4$), all have a short terminating representation in duodecimal (0.6, 0.4, 0.8, 0.3 and 0.9, respectively), and twelve is the smallest radix with this feature (since it is the least common multiple of 3 and 4). This all makes it a more convenient number system for computing fractions than most other number systems in common use, such as the decimal, vigesimal, binary, octal and hexadecimal systems.

Historically, units of time in many civilizations are duodecimal. There are twelve signs of the zodiac, twelve months in a year, and twelve European hours in a day or night. Traditional Chinese calendars, clocks, and compasses are based on the twelve Earthly Branches.

Following above reasons, a factorial *(n!)* of integer has more tail zeros in duodecimal system than in decimal. You can see, $(10)_{10}=(3628800)_{10}$ has 2 tail zeros, but $(A)_{12}=(1270000)_{12}$ has 4.

Input

Each line has one decimal positive integer n ($n < 10,000$). Zero means the end of input.

Output

For each input *n*, print the number of tail zeros of *n!* in one line.

Sample Input

```
4
10
2010
0
```

Sample Output

```
1
4
1001
```

Problem J: Buying and Selling

Mr. Skywind has received a request from his friend Lesley, who is a stock analyst. Lesley is focusing on the buying and selling quantity now. She will found a mathematics model on these values. As you know, every day, a lot of people sell stocks, and a lot of people buy. The quantities of stocks which are concluded are equal certainly. But the quantity that people want to sell and the value that people want to buy is not same. If buying quantity is more than selling, the difference indicated by a positive number is call net buying. Opposite, if the buying is small, the number is expressed using a negative number. So it can be called net-buying-selling value.

Although the difference is recoded every day, what Lesley focus is not only the daily value but also the sum of continuous days.

When a series of history stock data is given, Lesley wants to know that in which days the sum of net-buying-selling is maximum and positive or the sum is minimal and negative. Although Lesley can do some easy programming, it is hard to find an effective algorithm for a lot of data. Now it is the time that you write this program to help her.

Input

There are several test cases in input file. The first line of each case is an integer *n* *(2<=n<=100,000)* followed by *n* numbers which are less than 10,000. The end of input is indicated by a zero.

Output

For each case, two numbers are needed. The first is the maximum and positive sum of net-buying-selling in some continuous days; the second is the minimal and negative. You can assure there is one positive and one negative value at least.

Sample Input

```
6
1 2 3 -3 -2 -5
0
```

Sample Output

```
6 -10
```