Problem A: Funny plate (blue balloon)

Some burkinabe car registration numbers are written in the form of AX CD EYGH BF where A, X, E, Y, G, H are digits and C, D are letters. For example, 11 JN 3300 BF and 23 YV 7847 BF are burkinabe car registration numbers. The plate is called funny if AX can be read in EYGH. For example, the plate with a car registration number 11 ZY 1120 BF is a funny plate whereas the plate with a car registration number 23 YV 7847 BF is not a funny plate.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq$ **P** ≤ 25000), which is the number of data sets that follow. Each data set consists of one line containing a single registration number in the form of AX CD EYGH BF. There will be a single space between AX and CD, between CD and EYGH, and between EYGH and BF.

Standard Output

For each registration number, your program should display on one line, “funny plate” if a plate is a funny plate and “not a funny plate” if not.

Sample Input	Sample Output

`4`
`11 ZY 1120 BF`
`23 YV 7847 BF`
`09 CD 1092 BF`
`12 AB 0123 BF`
	`funny plate`
	`not a funny plate`
	`funny plate`
	`funny plate`

Problem B: Beautiful square (red balloon)

A beautiful square is a mirroring square of N rows and N columns (1 ≤ **N** ≤ 30) of integers between 1 and N^2 and which remains the same, after turning it on the left and after turning it on the top. N is the length of the square.

For example, consider the square in figure 1 that has a length of 3. After turning on the left, the initial square becomes the square in figure 2. After turning on the top, the initial square becomes the square in figure 3. The square in figure 1 is not a beautiful square.

1 2 3	3 2 1	7 8 9
4 5 6	6 5 4	4 5 6
7 8 9	9 8 7	1 2 3
Figure 1	Figure 2	Figure 3

The square in figure 4 is a beautiful square.

```
1 5 5 1
2 6 6 2
2 6 6 2
1 5 5 1
```

Figure 4

Standard Input

The first line of input contains a single integer **P**, (1 ≤ **P** ≤ 1000), which is the number of data sets that follow. Each data set consists of one line containing the length **N** of the square, followed by **N** lines containing the rows of the square.

Standard Output

For each data set, generate one line of output stating “beautiful square” or “not a beautiful square”, as shown in the sample below.

Sample Input	Sample Output
3 3 1 2 3 4 5 6 7 8 9 4 1 5 5 1 2 6 6 2 2 6 6 2 1 5 5 1 5 14 25 1 2 3 16 9 10 11 20 8 15 7 4 5 12 13 14 15 24 17 21 22 23 21	not a beautiful square beautiful square not a beautiful square

Problem C: Ouagadougou (yellow balloon)

The name Ouagadougou dates back to the 15th century when the Ninsi tribes inhabited the area. They were in constant conflict until 1441 when Wubri, a Yonyonse hero and an important figure in Burkina Faso's history, led his tribe to victory. He then renamed the area from "Kumbee-Tenga", as the Ninsi had called it, to "Wage sabre soba koumbem tenga", meaning "head war chief's village". Ouagadougou is a Francophone spelling of the name.
The city's name is often shortened to Ouaga.

Given a text, your task is to shorten Ouagadougou to Ouaga. In this problem, the name of the city OUAGADOUGOU will be exactly written in the input text as “Ouagadougou”.

Standard Input

The input is the text consisting of one or more lines, followed by a line containing only # that signals the end of the input. Each line will contain no more than 500 characters.

Standard Output

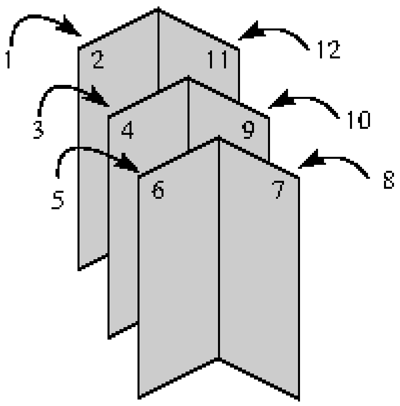
The output must be the same text with the word Ouagadougou shortened to Ouaga.

Sample Input	Sample Output
Ouagadougou is a beautiful city to visit. One day, i will travel to Ouaga-dougou city. #	Ouaga is a beautiful city to visit. One day, i will travel to Ouaga-dougou city.

Problem D: newspapers (green balloon)

Long ago, there were periodicals called *newspapers*, and these newspapers were printed on *paper*, and people used to *read* them, and perhaps even share them. One unfortunate thing about this form of media is that every so often, someone would like an article so much, they would take it with them, leaving the rest of the newspaper behind for others to enjoy. Unfortunately, because of the way that paper was folded, not only would the page with that article be gone, so would the page on the reverse side and also two other pages that were physically on the same sheet of folded paper.

For this problem we assume the classic approach is used for folding paper to make a booklet that has a number of pages that is a multiple of four. As an example, a newspaper with 12 pages would be made of three sheets of paper (see figure below). One sheet would have pages 1 and 12 printed on one side, and pages 2 and 11 printed on the other. Another piece of paper would have pages 3 and 10 printed on one side and 4 and 9 printed on the other. The third sheet would have pages 5, 6, 7, and 8.



When one numbered page is taken from the newspaper, the question is what other pages disappear.

Standard Input

Each test case will be described with two integers N and P, on a line, where $4 \leq N \leq 1000$ is a multiple of four that designates the length of the newspaper in terms of numbered pages, and $1 \leq P \leq N$ is a page that has been taken. The end of the input is designated by a line containing only the value 0.

Standard Output

For each case, output, in increasing order, the page numbers for the other three pages that will be missing.

Sample Input	Sample Output
12 2 12 9 8 3 0	1 11 12 3 4 10 4 5 6

Source: MCPC2013

Problem E : Root
(white balloon)

The digital root of a positive integer is found by summing the digits of the integer. If the resulting value is a single digit then that digit is the digital root. If the resulting value contains two or more digits, those digits are summed and the process is repeated. This is continued as long as necessary to obtain a single digit.

For example, consider the positive integer 24. Adding the 2 and the 4 yields a value of 6. Since 6 is a single digit, 6 is the digital root of 24. Now consider the positive integer 39. Adding the 3 and the 9 yields 12. Since 12 is not a single digit, the process must be repeated. Adding the 1 and the 2 yields 3, a single digit and also the digital root of 39.

Standard Input

The input file will contain a list of positive integers, one per line. The end of the input will be indicated by an integer value of zero.

Standard Output

For each integer in the input, output its digital root on a separate line of the output.

Sample Input	Sample Output
24 39 0	6 3

Source: GreaterNY2000

Problem F : Sums

(orange balloon)

For this problem you will compute various running sums of values for positive integers.

Standard Input

The first line of input contains a single integer **P**, (1<= **N** <= 10000), which is the number of data sets that follow. Each data set should be processed identically and independently. Each data set consists of a single line of input. It contains the data set number, **K**, followed by an integer **N**, (1<= **N** <= 10000).

Standard Output

For each data set there is one line of output. The single output line consists of the data set number, **K**, followed by a single space followed by three space separated integers **S1**, **S2** and **S3** such that:

- S1 is the sum of the first N positive integers.
- S2 is the sum of the first N odd integers.
- S3 is the sum of the first N even integers.

Sample Input

```
3
1
1
2 10
3 1001
```

Sample Output

```
1 1 1 2
2 55 100 110
3 501501 1002001 1003002
```

Source: GreaterNY2015

Problem G: Fingerpainting

(black balloon)

The local toy store sells small fingerpainting kits with between three and twelve 50ml bottles of paint, each a different color. The paints are bright and fun to work with, and have the useful property that if you mix X ml each of any three different colors, you get X ml of gray. (The paints are thick and "airy", almost like cake frosting, and when you mix them together the volume doesn't increase, the paint just gets more dense.) None of the individual colors are gray; the only way to get gray is by mixing exactly three distinct colors, but it doesn't matter which three. Your friend Emily is an elementary school teacher and every Friday she does a fingerpainting project with her class. Given the number of different colors needed, the amount of each color, and the amount of gray, your job is to calculate the number of kits needed for her class.

Standard Input

The input consists of one or more test cases, followed by a line containing only zero that signals the end of the input. Each test case consists of a single line of five or more integers, which are separated by a space. The first integer **N** is the number of different colors ($3 \le$ **N** ≤ 12). Following that are **N** different nonnegative integers, each at most 1,000, that specify the amount of each color needed. Last is a nonnegative integer **G** $\le$ 1,000 that specifies the amount of gray needed. All quantities are in ml.

Standard Output

For each test case, output the smallest number of fingerpainting kits sufficient to provide the required amounts of all the colors and gray. Note that all grays are considered equal, so in order to find the minimum number of kits for a test case you may need to make grays using different combinations of three distinct colors.

Sample Input	Sample Output
3 40 95 21 0 7 25 60 400 250 0 60 0 500 4 90 95 75 95 10 4 90 95 75 95 11 5 0 0 0 0 0 333 0	2 8 2 3 4

Source: MCPC2005