GLOBUSSOFT
TECHNOLOGY AHEAD OF TIME

Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION - 1

Alice and Bob need to send secret messages to each other and are discussing ways to encode their messages:

Alice: “Let’s just use a very simple code: We’ll assign ‘A’ the code word 1, ‘B’ will be 2, and so on down to ‘Z’ being assigned 26.”

Bob: “That’s a stupid code, Alice. Suppose I send you the word ‘BEAN’ encoded as 25114. You could decode that in many different ways!”

Alice: “Sure you could, but what words would you get? Other than ‘BEAN’, you’d get ‘BEAAD’, ‘YAAD’, ‘YAN’, ‘YKD’ and ‘BEKD’. I think you would be able to figure out the correct decoding. And why would you send me the word ‘BEAN’ anyway?”

Bob: “OK, maybe that’s a bad example, but I bet you that if you got a string of length 5000 there would be tons of different decodings and with that many you would find at least two different ones that would make sense.”

Alice: “How many different decodings?”

Bob: “Jillions!”

For some reason, Alice is still unconvinced by Bob’s argument, so she requires a program that will determine how many decodings there can be for a given string using her code.

Input

Input will consist of multiple input sets. Each set will consist of a single line of at most 5000 digits representing a valid encryption (for example, no line will begin with a 0). There will be no spaces between the digits. An input line of ‘0’ will terminate the input and should not be processed.

Output

For each input set, output the number of possible decodings for the input string. All answers will be within the range of a 64 bit signed integer.

Example

Input:

```
25114
1111111111
3333333333
0
```

Output:

```
6
89
1
```

QUESTION – 2

You are given a positive integer N. Let us consider set A of fractions x/y where 0 <= x/y <= 1, y <= N and the maximum common divisor of x and y is 1.

For example N = 5. Set A in increasing order consists of elements 0/1; 1/5; 1/4; 1/3; 2/5; 1/2; 3/5; 2/3; 3/4; 4/5; 1/1.

Your task is to find the i-th smallest fraction in set A.

Input

The first line of input contains the number of testcases t (t <= 15). The first line of each testcase contains numbers N and M (N <= 5000, M <= 10000). The next M lines contain one question each.

Output

For each testcase, you should output M lines which are the answers to the M questions.

Example

Input:

```
1
5 4
1
3
5
8
```

Output:

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

QUESTION – 3

Jill likes to ride her bicycle, but since the pretty city of Greenhills where she lives has grown, Jill often uses the excellent public bus system for part of her journey. She has a folding bicycle which she carries with her when she uses the bus for the first part of her trip. When the bus reaches some pleasant part of the city, Jill gets off and rides her bicycle. She follows the bus route until she reaches her destination or she comes to a part of the city she does not like. In the latter event she will board the bus to finish her trip.

Through years of experience, Jill has rated each road on an integer scale of niceness. Positive niceness values indicate roads Jill likes; negative values are used for roads she does not like. There are not zero values. Jill plans where to leave the bus and start bicycling, as well as where to stop bicycling and re-join the bus, so that the sum of niceness values of the roads she bicycles on is maximized. This means that she will sometimes cycle along a road she does not like, provided that it joins up two other parts of her journey involving roads she likes enough to compensate. It may be that no part of the route is suitable for cycling so that Jill takes the bus for its entire route. Conversely, it may be that the whole route is so nice Jill will not use the bus at all.

Since there are many different bus routes, each with several stops at which Jill could leave or enter the bus, she feels that a computer program could help her identify the best part to cycle for each bus route.

Input

The input file contains information on several bus routes. The first line of the file is a single integer *b* representing the number of route descriptions in the file. The identifier for each route (*r*) is the sequence number within the data file, $1 \leq r \leq b$. Each route description begins with the number of stops on the route: an integer *s*, $2 \leq s \leq 100000$ on a line by itself. The number of stops is followed by *s* - 1 lines, each line *i* ($1 \leq i < s$) is an integer n_i with absolute value ≤ 1000 representing Jill's assessment of the niceness of the road between the two stops *i* and *i*+1.

Output

For each route in the input file, your program should identify the beginning bus stop *i* and the ending bus stop *j* that identify the segment of the route which yields the maximal sum of niceness, $m= n_i+n_{i+1}+...+n_{j-1}$. If more than one segment is maximally nice, choose the one with the longest cycle ride (largest *j*-*i*). To break ties in longest maximal segments, choose the segment that begins with the earliest stop (lowest *i*). For each route *r* in the input file, print a line in the form:

```
The nicest part of route r is between stops i and j
```

However, if the maximal sum is not positive, your program should print:

```
Route r has no nice parts
```

Example

Input:

```
3
3
-1
6
10
4
-5
4
-3
4
4
-4
4
-5
4
-2
-3
-4
```

Output:

```
The nicest part of route 1 is between stops 2 and 3
The nicest part of route 2 is between stops 3 and 9
Route 3 has no nice parts
```

QUESTION – 4

In a billiard table with horizontal side **a** inches and vertical side **b** inches, a ball is launched from the middle of the table. After **s** > 0 seconds the ball returns to the point from which it was launched, after having made **m** bounces off the vertical sides and **n** bounces off the horizontal sides of the table. Find the launching angle **A** (measured from the horizontal), which will be between 0 and 90 degrees inclusive, and the initial velocity of the ball.

Assume that the collisions with a side are elastic (no energy loss), and thus the velocity component of the ball parallel to each side remains unchanged. Also, assume the ball has a radius of zero. Remember that, unlike pool tables, billiard tables have no pockets.

Input

Input consists of a sequence of lines, each containing five nonnegative integers separated by whitespace. The five numbers are: **a**, **b**, **s**, **m**, and **n**, respectively. All numbers are positive integers not greater than 10000.

Input is terminated by a line containing five zeroes.

Output

For each input line except the last, output a line containing two real numbers (accurate to two decimal places) separated by a single space. The first number is the measure of the angle **A** in degrees and the second is the velocity of the ball measured in inches per second, according to the description above.

Example

Input:

```
100 100 1 1 1
200 100 5 3 4
201 132 48 1900 156
0 0 0 0 0
```

Output:

```
45.00 141.42
33.69 144.22
3.09 7967.81
```

QUESTION – 5

There is a string containing only decimal digit characters. The length of the string is between *1* and *1000*. Using characters of the string, you have to construct the maximum number which divides by fifteen without remainder. Each character of the string may not be used more than once.

Input

First line of input contains an integer **t** (*1* ≤ **t** ≤ *90*), equal to the number of testcases. Then descriptions of **t** testcases follow.

Each testcase is described in a single line representing the source string.

Output

For each testcase output one line with the decimal representation of the maximum number. Leading zeroes should be omitted. If no number can be constructed, output a single word *“impossible”*.

Example

`Input:`

```
1
02041
```

`Output:`

```
4200
```