Problem A. Buses

Input file:	`buses.in`
Output file:	`buses.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Vasya works for the RIBA (Research Institute of Buses Automatization). He is studying the bus routes system in the city of Nsk.

There are n bus routes in Nsk, numbered 1 through n. Moreover, the numbers of routes have the following property: the length of the common part of routes a and b is equal (in bus stops) to the length of the longest common substring of decimal representation of a and b.

Help Vasya to calculate $\sum_{a=1}^{n} \sum_{b=1}^{n} f(a, b)$, where $f(a, b)$ is length of common parts of routes a and b.

Input

The input file contains not more than 10 test cases. Each test case consists of one line containing one integer number n ($1 \le n \le 10^5$). The sum of all n-s in the input file does not exceed 10^5.

Output

For each test case write the only integer — the answer to the problem. Adhere to the sample output format below as close as possible.

Example

buses.in	buses.out
17 30 239	Case #1: The sum of lengths is 109. Case #2: The sum of lengths is 367. Case #3: The sum of lengths is 34815.

Problem B. Count Primes

Input file:	`cntprime.in`
Output file:	`cntprime.out`
Time limit:	20 seconds
Memory limit:	256 Mebibytes

Research Institute of Primes (R.I.P.) has a new task for Vasya. He should write a program which finds how many primes are located on a given segment $[L, R]$.

Input

First line of the input file contains a single integer T — the number of test cases ($1 \le T \le 1000$). Each test case is described by a single line containing two positive integers L_k and R_k — the bounds of the segment ($1 \le L_k \le R_k$).

$$\sum_{k=1}^{T}(R_k + L_k) \le 10^{12} + 1$$

Output

For each test case write the only integer formatted as shown below — the answer to the problem. Adhere to the sample output format as close as possible.

Example

`cntprime.in`	`cntprime.out`

`2`
`1 1`
`2 3`
	`Case #1: Segment contains 0 prime(s).`
	`Case #2: Segment contains 2 prime(s).`

Problem C. Epidemic

Input file:	`epidemic.in`
Output file:	`epidemic.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Research Institute of Grippe (R.I.G.) has a task for Vasya. Vasya should simulate epidemic of grippe. The mathematical model of epidemic is described below.

N people live in a small village. At day 0, one of them was infected (we denote this person by number 1) and others were healthy. A person infected at day d stays ill for T successive days (these are days d, $d+1$, $d+2$, $\ldots$, $d+T-1$). After that, that person is again healthy and gains immunity to the epidemic. For example, person number 1 will be healthy at day T and can't be infected again after that.

If a healthy person without immunity meets an infected person at day d, this healthy person is infected at that day. There is one exception, however: no one spreads infection on the first day of illness. For example, if a healthy person A becomes infected and then meets another person B on the same day, that person B does not become infected because of this meeting.

You are given the probabilites of people meeting each other on each day; the probability that person i meets person j during any single day is equal to p_{ij}.

Help Vasya to write a program which finds the average number of unhealthy people during the first K days.

Input

The first line of the input file contains a single positive integer — the number of test cases. The first line of each test case contains three positive integers N, K and T ($2 \leq N$, K, $T \leq 200$). Next N lines contain the matrix P of meeting probabilities. It is guaranteed that $0 \leq p_{ij} \leq 1$, $p_{ij} = p_{ji}$ and $p_{ii} = 0$. Every probability p_{ij} given with 3 digits after decimal point. The sum of all N in input doesn't exceed 1000.

Output

For each test case and for each day, you should output the average number of unhealthy people during that day. Output these numbers as precisely as possible. Absolute error of your answer shouldn't be greater than 10^{-6}. Adhere to the sample output format below as close as possible.

Example

`epidemic.in`

```
1
3 3 2
0.000 0.100 0.200
0.100 0.000 0.300
0.200 0.300 0.000
```

`epidemic.out`

```
Case #1:
  Day 1, average number of unhealthy people equals 1.3
  Day 2, average number of unhealthy people equals 0.378
  Day 3, average number of unhealthy people equals 0.0966624
```

Problem D. Grass

Input file:	`grass.in`
Output file:	`grass.out`
Time limit:	40 seconds
Memory limit:	256 Mebibytes

Vasya works for the RIGHT (Research Institute of Grass, Herbs and Trava). He has an excellent experimental lawn that has a form of a convex polygon. There's also a stake on the lawn (or on its border): Vasya's pet, Petya the Goat, is tied to it.

Due to financial problems in the institute, Vasya is forced to give away some part of the lawn. Now he can afford only a triangular lawn with vertices coinciding with some of original ones. Certainly, the stake has to be inside or on the border of the new lawn.

Help Vasya to find the triangle of maximal area having the stake inside or on the border.

Input

The input file contains no more than 10 test cases. Each test case starts with one line containing one integer N — the number of vertices of the lawn ($3 \le N \le 10^4$). $N + 1$ lines follow, containing two coordinates each: vertices of the lawn in counter-clockwise order and the stake for the goat. All coodrinates do not exceed 10^8 by absolute value. The polygon is non-degenerate.

Output

For each test case write three different integers from 1 to N — the numbers of vertices. Adhere to the sample output format below as close as possible.

Example

`grass.in`

```
4
0 0
4 0
4 4
0 4
3 2
```

`grass.out`

```
Case #1: The maximal triangle is formed by vertices 1, 2 and 3.
```

Problem E. Hypercube

Input file:	`hypercube.in`
Output file:	`hypercube.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Vasya works for the RICH (Research Institute of Concrete (continual/discrete) Hyperspace). He is studying the following problem: given a Z-dimensional (Z is the day when Vasya was born, so it is his favorite number) hypercube, cut it into y pieces of equal shape and volume. After cutting the hypercube into x equal (coinciding after a shift) pieces, Vasya decided to limit himself with the following operation: select some subset of pieces A satisfying the following conditions

- union of all elements of A is a rectangular parallelepiped
- current set of pieces can be split into several subsets B_i such that:
  - The intersection of subsets $B_i \cap B_j = \varnothing$ for $i \neq j$
  - The set of points of $\bigcup\limits_i B_i$ forms the hypercube itself
  - Each B_i is equal to A up to parallel shift

and replace all current pieces with shrinked copies of A. Copies of A should be shrinked to the size of current pieces, no rotation is allowed. Shrinking is performed along each of the axes independently.

Determine the shortest sequence of such operations required to cut the hypercube into y equal pieces, or find out that it is impossible.

Input

The input file contains no more than 1000 test cases. Each test case consists of one line containing two integers x and y ($1 \leq x, y \leq 10^{18}$).

Output

For each test case, write any of the shortest sequences or report that there's no such sequence. Adhere to the sample output format below as close as possible.

Example

hypercube.in

`3 9`
`17 239`
`10 10000`

hypercube.out

`Case #1: The shortest sequence is: 3 -> 9`
`Case #2: No way`
`Case #3: The shortest sequence is: 10 -> 100 -> 10000`

Problem F. Lottery

Input file:	`lottery.in`
Output file:	`lottery.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Research Institute of Primes (R.I.P.) with cooperation with Research Institute of Stepping Knights (RISK) organizes an extraordinary lottery. Every participant of the lottery gets two 4×4 grids. The first grid is filled with decimal digits, the second one is empty. A participant should first choose an integer K ($1 \leq K \leq 16$) and then place natural numbers from 1 to K on the second grid. These numbers should correspond to steps of some path of chess knight on the grid: each number should be placed exactly once and there is one knight's step between each two consecutive numbers. Of course, knight couldn't visit the same cell twice.

After that, the participant writes down K digits from the first grid in order of traversing the knight's path from the second grid. These K digits form some integer N. If N is prime, the participant gets an amount of prize money proportional to N; otherwise, he doesn't get any money.

However, there's one small problem. The director of the lottery can't calculate maximum sum of all prize (prime) numbers of all printed tickets. Therefore, he doesn't know what should be the exchange rate from numbers to dollars. So he asked Vasya to write a program calculating the maximum prime number one could obtain from a given ticket.

Input

First line of the input file contains single integer T — the number of tickets in the input ($1 \leq T \leq 100$). The description of each ticket consists of a single line which contains 16 decimal digits. The digits are listed left to right, top to bottom.

Output

For each test case, write the only integer — the answer to the problem (without leading zeroes). Adhere to the sample output format below as close as possible.

Example

`lottery.in`

```
3
4444444444444444
1357975313579753
2222222222222222
```

`lottery.out`

```
Ticket 1 is unlucky.
Ticket 2, maximum prime is 977597533351357.
Ticket 3, maximum prime is 2.
```

Second ticket (example)

1	**3**	**5**	**7**
9	**7**	**5**	**3**
1	**3**	**5**	**7**
9	**7**	**5**	**3**

	13	**4**	**15**
5	**2**	**7**	**10**
12	**9**	**14**	**3**
1	**6**	**11**	**8**

Problem G. Poly

Input file:	`poly.in`
Output file:	`poly.out`
Time limit:	5 seconds
Memory limit:	256 Mebibytes

Vasya works for the RIP (Research Institute of Polynomials). He studies polynomials with integer coefficients. Recently, he read the method to find out whether the polynomial has rational roots. Now he wants to solve a more complicated problem — he wants to find whether the polynomial has quadratic-irrational roots — that is, roots which can be written in the form $x = a + \sqrt{b}$ where $a, b \in \mathbb{Q}$ and $b > 0$. Help him.

Input

The input file contains no more than 10 test cases. Each test case consists of one line. The first number will be an integer $0 \le n \le 100$ representing the degree of polynomial. Next $n + 1$ integers will represent coefficients of polynomial starting with coefficient by x^n. The coefficient by x^n will not be equal to zero. All coefficients will not exceed 10^3 by magnitude.

The sum of degrees for all tests will not exceed 100.

Output

For each test case, write the answer on a single line. Adhere to the sample output below. If the polynomial has more than one quadratic-irrational root, write any of them. Greatest common divisor of numerator and denominator of each fraction should be equal to 1.

Example

poly.in

`2 1 0 1`
`2 1 0 -100`
`2 1 2 -1`

poly.out

`Case #1: Polynomial does not have such roots`
`Case #2: Polynomial has rational root 10/1`
`Case #3: Polynomial has root -1/1+sqrt(2/1)`

Problem H. Sociology

Input file:	`sociology.in`
Output file:	`sociology.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Vasya works for the RISK (Research Institute of Sociological tasKs). He is studying relationships between software engineers working freelance and their managers. Vasya considers several jobs assigned to programmers. Each job is to be done by one engineer and is to be managed by one manager.

Vasya calls a subset A of managers *excessive* if the subset of engineers having common jobs with at least one of managers from A has lesser cardinality than $|A|$. His hypothesis is that a system having no excessive subsets of managers is more stable and produces less pressure to the workers.

Your task is to find an excessive subset or say that it is impossible.

Input

Input consists of no more than 10 test cases. The first line of each test case contains two integers N_e and N_m — the number of engineers and managers, respectively ($1 \leq N_e, N_m \leq 10^4$). The next line contains a single integer N_j — the number of jobs ($1 \leq N_j \leq 10^5$). Then N_j lines follow, each containing two integers e_i and m_i — numbers of engineer and manager assigned to job number i.

Output

For each test case, output either an excessive subset of managers or a message that it does not exist. Adhere to the sample output below as close as possible.

Example

sociology.in

```
3 3
6
1 2
1 3
2 3
2 1
3 1
3 2
2 3
2
1 3
2 2
1 3
3
1 1
1 2
1 3
```

sociology.out

```
Case #1: There is no excessive subset of managers.
Case #2: Manager 1 forms an excessive subset.
Case #3: Managers 1, 2 and 3 form an excessive subset.
```

Problem I. Stones Again!

Input file:	`stones.in`
Output file:	`stones.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Research Institute of Geometry (RIG) has a task for Vasya. He must put N stones on the board for "Go" game. The coordinates of stones' positions should be integers from $-N$ to N inclusive. Vasya can't put two stones into the same position. Moreover, no three stones can lie on a straight line. For example, Vasya couldn't place a stone in position $(4, 2)$ if he already placed stones at positions $(2, 1)$ and $(0, 0)$.

After placing stones on the board, Vasya must draw a simple polygon with N vertices by connecting stones with lines. A *simple polygon* is a closed polygonal chain of line segments on the plane which do not have common points other than the common vertices of pairs of consecutive segments. In particular, a vertex cannot lie strictly inside a side of the polygon. Note that the polygon does not need to be convex.

Help Vasya to solve this tricky problem.

Input

First line of the input file contains single integer T — the number of test cases ($1 \le T \le 100$). Each test case is described by single line containing one positive integer N ($3 \le N \le 1000$). The total sum of all Ns does not exceed 10 000.

Output

For each test case, write coordinates of polygon vertices formatted as shown below. Your could output vertices in clockwise or counter-clockwise order. In case of multiple solutions, any of them will do. Adhere to the sample output format below as close as possible.

Example

stones.in
2 3 4
stones.out
Case #1: (0, 0) - (3, 3) - (3, -3) Case #2: (0, 0) - (1, 1) - (0, -2) - (-1, 1)

Problem J. String Machine

Input file:	`turing.in`
Output file:	`turing.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Research Institute of Turing Machines (RITM) developed a strange string machine. This machine works with strings of decimal digits. For a given input string A, machine could either produce string B or hang. If for a given input string A, machine produced string B, we will write that string A produces string B. There are two production rules (here, XY denotes concatenation of strings X and Y):

Rule 1: String $2S$ produces string S.

Rule 2: If string X produces string Y, string $3X$ produces string $Y2Y$.

S, X, Y are arbitrary non-empty strings of decimal digits. Rules can be applied recursively. For example, string 327 produces string 727 and string 3327 produces string 7272727. If no rule could be applied to the given string, the string machine will hang.

Vasya conducts the following experiment. As a first iteration, he sends string A to the input of the string machine. If the machine hangs, the experiment is over. Otherwise, he takes the output and sends it again to the input, thus performing the second iteration, and so on, ten iterations in total. Your task is to find out the number of iteration at which the machine hangs, if any.

Input

First line of the input file contains single integer T — the number of test cases ($1 \le T \le 100$). Each test case is described by single line containing string A of decimal digits ($1 \le |A| \le 100$).

Output

For each test case, write the number of iteration at which the string machine hangs. If machine doesn't hang after ten iterations, write the corresponding message. Adhere to the sample output format below as close as possible.

Example

turing.in
6 2 327 221 2221 323 2222222222
turing.out
Case #1: Machine hangs at 1st iteration. Case #2: Machine hangs at 2nd iteration. Case #3: Machine hangs at 3rd iteration. Case #4: Machine hangs at 4th iteration. Case #5: Machine doesn’t hang at 10th iteration. Case #6: Machine hangs at 10th iteration.