Problems Overview

Problem A: Digit Counting

Problem B: Numerical surprises

Problem C: Prime k-tuple

Problem D: The longest constant gene

Problem E: Lazy Susan

Problem F: Square Painting

Problem G: The Ultimate Password

Problem H: Plagiarism Detection

Problem I: Space Beacon

Problem J: Ig-Nobel Prize

Note: The input and output for all the problems are standard input and output.

Problem A: Digit Counting

Trung is bored with his mathematics homeworks. He takes a piece of chalk and starts writing a sequence of consecutive integers starting with 1 to N (1<N<10000). After that, he counts the number of times each digit (0 to 9) appears in the sequence. For example, with N = 13, the sequence is:

12345678910111213

In this sequence, 0 appears once, 1 appears 6 times, 2 appears 2 times, 3 appears 3 times, and each digit from 4 to 9 appears once.

After playing for a while, Trung gets bored again. He now wants to write a program to do this for him. Your task is to help him with writing this program.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each test case, there is one single line containing the number N.

Output

For each test case, write sequentially in one line the number of digit 0, 1, …9 separated by a space.

Sample Input	Sample Output
2 3 13	0 1 1 1 0 0 0 0 0 0 1 6 2 2 1 1 1 1 1 1

Problem B: Numerical surprises

We suspect that for every positive integer *N* there exists an integer of the form 11…10…0 (a sequence of 1's followed by 0 or more 0's) that is divisible by *N*. For example, with *N* = 3, 111 is divisible by 3, with *N* =4, 100 is divisible by 4, with *N* = 7, 11111 is divisible by 7. We want to verify this for some integers. The solution to this problem is to find two different numbers *P* and *Q* in the form of 11…1 (a sequence of 1's) that have the same remainder when dividing by *N*. The difference *D* between *P* and *Q* will be in the form of 11…10…0 and divisible by *N*.

In order to solve this problem, we have to start with finding the remainder when dividing a number in the form of 11…1 by *N*. Your task is to write a program to do this.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

Each data set is described by two lines. The first line contains the integer *N* ($1 < N < 10^9$). The second line contains the integer number *P* (*P* contains at least one digit and at most 1000 digits).

Output

For each test case, write in one line the remainder when dividing *P* by *N*.

Sample Input	Sample Output
2 4 11 5 111	3 1

Problem C: Prime k-tuple

$\{p_1,..., p_k : p_1 < p_2 < ... < p_k\}$ is called a prime k-tuple of distance s if $p_1, p_2, \ldots, p_k$ are consecutive prime numbers and $p_k - p_1 = s$. For example, with $k = 4$, $s = 8$, {11, 13, 17, 19} is a prime 4-tuple of distance 8.

Given an interval [a, b], k, and s, your task is to write a program to find the number of prime k-tuples of distance s in the interval [a, b].

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each data set, there is only one line containing 4 numbers, a, b, k and s ($a, b < 2*10^9$, $k < 10$, $s < 40$).

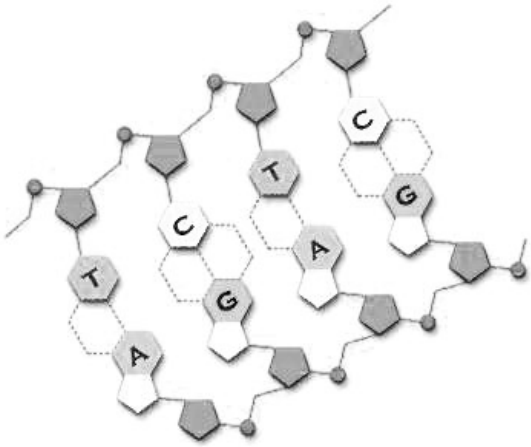
Output

For each test case, write in one line the numbers of prime k-tuples of distance s.

Sample Input	Sample Output
1 100 200 4 8	2

Problem D: The longest constant gene

Thousands of genomes, from virus to human, are available in public databases. Each genome is presented as a string of nucleotides: "A", "C", "G", and "T". To study the relationship among organisms, their genomes are analyzed. A sequence of nucleotides is called a *constant gene* if it appears in all genomes.



Given N genomes, your task is to write a program to find the longest constant gene among these genomes.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each data set, the first line contains the integer N ($1 < N < 7$) indicating the number of genomes. Each line in the next N following lines contains one genome (the length of each genome is limited to one million).

Output

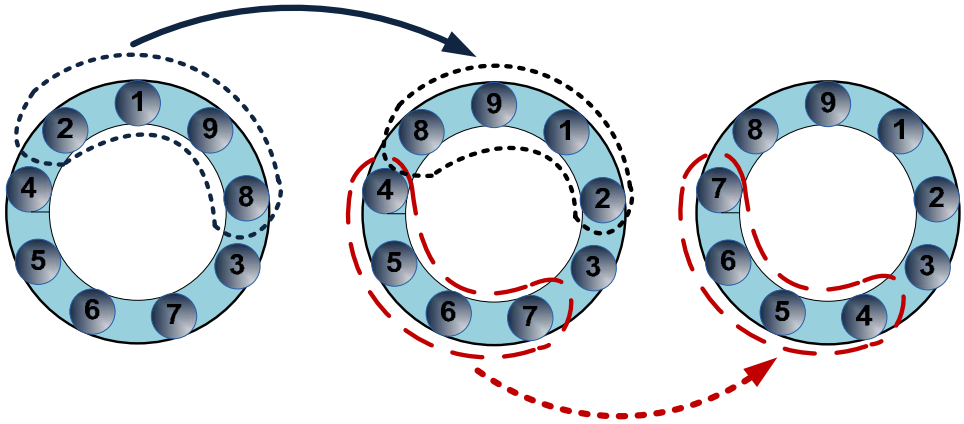
For each test case, write in one line an integer number indicating the length of the longest constant genome.

Sample Input	Sample Output
2 2 ACGGG**CGTCGTCCCCGTCGTCGT**ATC CT**CGTCGTCCCCGTCGTCGT**GTC 3 ACG**ACGGCTGCGGT**AACCC TT**ACGGCTGCGGT**CCCCTT CCCCCCGTTT**ACGGCTGCGGT**GG	18 11

Problem E: Lazy Susan

There are *N* marbles, which are labeled 1, 2, …, *N*. The *N* marbles are put in a circular track in an arbitrary order. In the top part of the track there is a "lazy Susan", which is a tray that can hold exactly 4 marbles. The tray can be rotated, reversing the orientation of the four marbles. The tray can also be moved around the track in both directions.

For example, 9 marbles 1, 9, 8, 3, 7, 6, 5, 4, 2 are put in the circular track in clockwise order as shown in the following figure. This figure also shows how the tray is moved and rotated.



Trung wants you to arrange the marbles by moving and rotating the tray so that when listing the marbles from some position in the track in clockwise order, we get (1, 2…, *N*). Your task is to write a program to tell Trung that either this can be done or not.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 100. The following lines describe the data sets.

For each data set, the first line contains the integer N ($8 \le N \le 500$). The second line describes the initial state of the track. It contains N numbers which are the labels of the marbles when listing in clockwise order.

Output

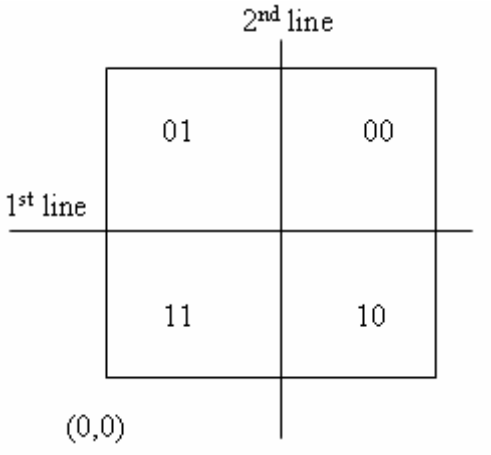
For each test case, write in one line "**possible**" if there exists a solution to arrange the marbles. If not so, write "**impossible**".

Sample Input	Sample Output
2 9 1 9 8 3 7 6 5 4 2 11 1 3 2 4 5 6 7 8 9 10 11	possible impossible

Problem F: Square Painting

Given a 10000×10000-square in the Cartesian plane with 4 corners (0,0) (0,10000) (10000,10000) (10000,0). We consecutively draw N lines, which are numbered from 1 to N. These lines subdivide the given square into M pieces, of which all edges are longer than 1. Suppose that each line intersects exactly two edges of the given square and each line does not pass any square corner.

The i^{th} line divides the square into two parts: P_i containing (0,0), and Q_i not containing (0, 0). After all the subdivisions, each resulted piece is labeled by a nonnegative integer, which has binary representation $b_1b_2...b_n$, where b_i =1 if the piece lies in P_i, and b_i = 0 if the piece lies in Q_i. For example, the two lines in the figure subdivide the square into 4 pieces.



Now they want to paint all the pieces with S colors denoted by integers from 1 to S such that:

- Each piece is painted with only one color;
- Two adjacent pieces (having a common edge) must be painted with two different colors.
- The piece with the smallest label must be painted with color 1.

Your task is to write a program that determines the solution to paint the pieces with the minimum number of colors S.

Input

The input consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each data set, the first line contains a single positive integer N ($N \le 30$) – the number of lines. The i^{th} line of the following N lines contains 4 ***integers*** x_i, y_i, z_i, t_i - the coordinates of the intersections of i^{th} line with two square edges.

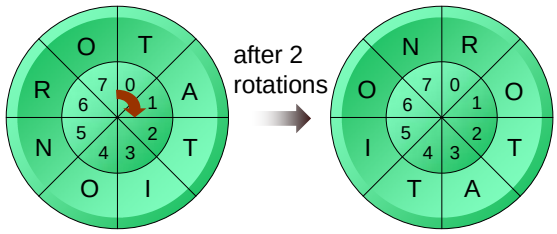
Output

For each test case, output the solution in a number of lines. In the first line, write the minimum number of colors S needed for painting. The next M lines described how each piece is painted in increasing order of the label of all the pieces. In each line, write two numbers separated by a space, the former is the label of the piece and the latter is the color used to paint that piece.

Sample Input	Sample Output
2 1 0 5000 10000 5000 2 0 5000 10000 5000 5000 0 5000 10000	2 0 1 1 2 2 0 1 1 2 2 2 3 1

Problem G: The Ultimate Password

A "*letter*" lock is a circle, in which we mark some positions equidistant from one another. The positions are numbered clockwise by a zero-based index and there is exactly one English capital letter put in each position. The state of the lock is given by a string, which contains all letters enumerated from position 0 to the end. The lock can change its state by performing rotations: After a rotation, every letter in the circle will move clockwise to the next position. If the lock received a password in form of a string, it will step by step rotate to change its state and verify whether its current state matches the prefix of the password. Whenever it matches, the lock will be unlocked, and the verification process will stop successfully. If the lock moves around and there has been no matched state found, it will delete the first character in the password and retry with the new password, and so on. This process repeats until the lock is unlocked or the password has been completely deleted (verification fails).



Given a sequence of *N* locks, in which the state of any lock is not shorter than the state of its previous lock, one may want to unlock all of them in succession by only one password. The process is as follows: the password is first applied to the first lock in the sequence; the remaining password after unlocking the first lock will be applied again to the second lock, the remaining password after unlocking the second lock will be applied again to the third lock, etc. The process continues until the last lock is unlocked.

Your task is to write a program to find the shortest password (in terms of the number of characters) that can be used to unlock all these *N* locks in succession. If there is more than one solution, just find the first one in lexicographic order.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each data set, there is only one line containing a string that lists all states of the locks separated by one semicolon (";"). In each test case, the number of locks is not greater than 200, the state of every lock is not empty and consists of less than 101 letters.

Output

For each test case, write in one line the shortest password to unlock all the locks in succession.

Sample Input	Sample Output
3 TOPO;POFTTO;THEPOF;HEWOOFT;HEWORLDT WELC;COMEEL;METLCO;TOCOME HEACMT;PROGRAMCM;RAMMINGCON;CONTESTING	TOPOFTHEWORLD WELCOMETO THEACMPROGRAMMINGCONTEST

Problem H: Plagiarism Detection

A local school has noticed an increasing number of cases where electronic essays submitted by students are very similar. The teachers come up with an effective method to automatically detect similar documents using ***n***-grams as the building block. An ***n***-gram is a sequence of ***n*** consecutive words. Two ***n***-grams are considered equal if their contents are exactly the same (case sensitive).

A word is a sequence of consecutive characters delimited by one or more whitespaces, punctuations or boundaries of email addresses. A whitespace can take the following ASCII codes: 9 (horizontal tab), 10 (new line), 11 (vertical tab), 12 (new page), 13 (carriage return) or 32 (space). A punctuation character can be any character with the following ASCII codes: 33 (exclamation mark), 44 (comma), 46 (period), 58 (colon), 59 (semicolon) and 63 (question mark). But when the period is part of an email address, the whole email address is considered to be a word.

An email address has the format: ***local-part@domain-part***. Both the local part and the domain part may only contain the following ASCII characters:

- Uppercase and lowercase letters
- The digits 0 through 9
- The characters with ASCII codes 45(-) or 95 (_)
- The period (.) provided that it is not the first nor the last character in the local part or the domain part, nor may it appear two or more times consecutively.

Additionally, the domain part must contain at least one period and an email address is the longest string of characters satisfying the above-mentioned conditions.

For example, the following document:

Email address, Email address: admin.team@company.com.vn.

has three unique 2-grams namely: "*Email address*", "*address Email*" and "*address admin.team@company.com.vn*". Notice that this document has only one email address, which is *admin.team@company.com.vn*, and its substring *team@company.com* is not a valid email address.

For a given value of ***n***, suppose document d_1 has ***m*** unique ***n***-grams, document d_2 has ***l*** unique ***n***-grams and the two documents have ***s*** unique ***n***-grams in common. The similarity between two documents is calculated with the following formula:

$$similarity(d_1, d_2) = \min\left(\frac{s}{m}, \frac{s}{l}\right) * 100$$

Your task is to write a program to compute the similarity percentage between two given documents for a given value of ***n***.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each data set, the first line of each data set contains one integer ***n*** (1 <= ***n*** <= 10) where n denotes which ***n***-gram to use. The second line contains two integers ***k*** and ***l***, which are the number of lines the first and second documents contain respectively. The next ***k*** lines contain the content of the first document. The next ***l*** lines contain the content of the second document.

Output

For each test case, write in one line the similarity percentage, truncated to 2 decimal places, between the two input documents.

Sample Input	Sample Output
1 2 2 1 John, my friend, is a good programmer. My friend John is a good programmer!	50.00

Problem I: Space Beacon

In order to discover all the planets of the solar system, we want to develop techniques to travel safely through an asteroid belt between Mars and Jupiter. We plan to drop automatic-signaling devices into large asteroids of the belt, which will act as space beacons to guide the ships. They will assist autopilots to track the location of the ships to adjust the orbit. Each signal sent by each beacon contains a sequence of pulses, and is characterized by a sequence *T*:

$$T = t_1, t_2, \ldots, t_k, (3 \le k \le 18),$$

where t_i is the duration of i-th pulse (t_i is integer and is in the range of 1..9).

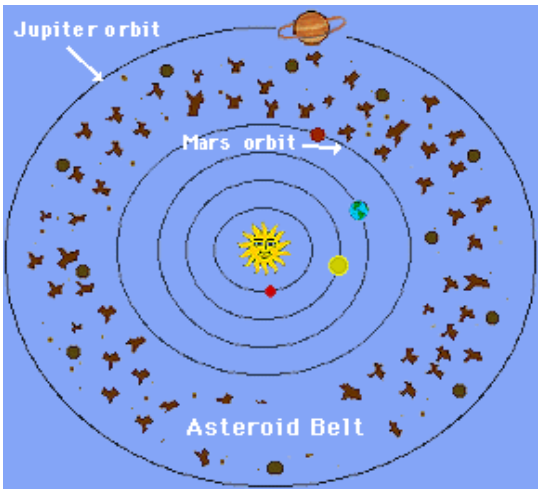
In order to simplify the technical checking process and to increase the signal recognition ability, the sequence *T* of each beacon is designed with the following criteria:

With $1 < i < k$, either:

$$\begin{cases} t_{i-1} < t_i, t_i > t_{i+1} & \text{for i mod } 2 = 0 \\ t_{i-1} > t_i, t_i < t_{i+1} & \text{for i mod } 2 = 1 \end{cases}$$

or

$$\begin{cases} t_{i-1} > t_i, t_i < t_{i+1} & \text{for i mod } 2 = 0 \\ t_{i-1} < t_i, t_i > t_{i+1} & \text{for i mod } 2 = 1 \end{cases}$$



All possible *T* sequences are sorted in lexicographic order and labeled by consecutive integers starting with 1. The label of the sequence *T* of each beacon is used as the identifier of the beacon.

Given the sequence *T* of a beacon, your task is to write a program to find the identifier of that beacon.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

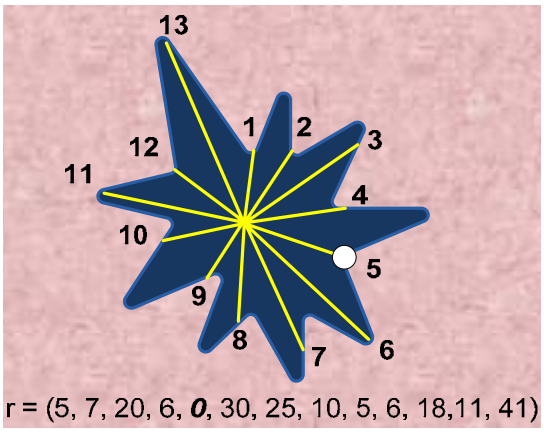
For each data set, there is only one single line containing *k* integers t_1, t_2..., t_k separated by space describing the *T* sequence of a beacon.

Output

For each test case, write in one line the ID of the beacon with the given *T* sequence.

Sample Input	Sample Output
2 1 2 1 2 1 2 1 2 1 2	2 4

Problem J: Ig-Nobel Prize



One of the scientific research projects was almost chosen to the shortlist of the Ig Nobel. It was rejected because the author stated that the analyzed data is not sufficient, and that it will be studied further in the future. The research is to investigate the shape of ink drops left by grade 1 students in their notebooks. The author chose a point in the ink drop to act as the center. A line starting from that point rotates 360^o; and whenever it hits an angle of $360^o/\boldsymbol{n}$, the intersection of the line and the border of the drop will be marked. At the i^{th} mark, we get the i^{th} intersection ($\boldsymbol{i} = 1, 2, \ldots, \boldsymbol{n}$). Thus, after the $(\boldsymbol{n}\text{-}1)^{th}$ is the $\boldsymbol{n}^{th}$, and after the $\boldsymbol{n}^{th}$ is the 1^{st} one. The author recorded the length r_i from the center to the i^{th} intersection by ($1 \le r_i \le 10^6$). However, at the m^{th} marking of the intersection, he accidentally made a hole in the paper. Therefore, he didn't have the exact r_m. With the honesty of a scientist, he left r_m blank, recorded as a value of 0. One of the research goals is to find a rain drop which has a shape closest to that of the ink drop. In order to find it, the author planned to eliminate a minimal number of intersections so that the $\boldsymbol{r_i}$ of the remaining intersections can form a **descending order** of values starting from a certain intersection by either directions (in a circular way). In this process, the unknown value $\boldsymbol{r_m}$ is assigned by a value $\boldsymbol{k}$ in the range from 1 to $\boldsymbol{n}$.

Given n and r_i ($\boldsymbol{i} = 1 \div \boldsymbol{n}$), your task is to write a program to find the value of $\boldsymbol{k}$ in order that the number of the eliminated intersections is minimal. In the case that there is more than one value of $\boldsymbol{k}$, find the smallest one.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each test case, the first line contains the integer $\boldsymbol{n}$ ($1 < \boldsymbol{n} \le 10\,000$). The second line contains n integers $r_1, r_2, \ldots, r_{\boldsymbol{n}}$ separated by space.

Output

For each test case, write in one line two integers: $\boldsymbol{k}$ and the minimal number of eliminated intersections separated by one single space.

Sample Input	Sample Output
1 5 5 7 0 7 4	1 1