Problems Overview

Problem A: Digit count

Problem B: Mysterious Treasure

Problem C: Prime Concatenation

Problem D: Bumps

Problem E: Name Split

Problem F: Alien Message

Problem G: Factorization

Problem H: Decoding

Problem I: Viruses

Problem J: Page Layout

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

Problem A: Digit count

Given an integer N ($1 < N < 10^9$), your task is to write a program to count the number of non-zero digits in N that N is divisible by.

For example, with $N = 1359$, N is divisible by 1, 3 and 9. So the result in this case is 3.

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 consists of a single line which contains a single integer number N.

Output

For each data set write in one line the number of digits as mentioned above.

Sample Input	Sample Output
3	**3**
1359	**2**
3520	**5**
33399	

Problem B: Mysterious Treasure

The planet of Nine Dragons is famous not only for its beautiful scene but also for the legend about the secret treasures. As an astronaut, you have found an old map guiding you to the secret cave where the treasures are hidden. Unfortunately, the door of the cave is locked and is only opened at certain times of the day.

A day in this planet is very long as it has 10^9 minutes. The door of the cave is opened at the n^{th} minutes (from midnight) if $n! = 1 \times 2 \times \ldots n$ is divisible by n^3 (i.e., $n! \bmod n^3 = 0$).

Given n, your task is to write a program to determine if the door of the cave is opened at the n^{th} minute.

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 consists of one line containing the integer n ($1 < n < 10^9$).

Output

For each data set write in one line "YES" if the door is opened at the given n^{th} minute or "NO" otherwise.

Sample Input	Sample Output
2	**NO**
6	**YES**
12	

Problem C: Prime Concatenation

Given two positive integers n and d, your task is to write a program to find the smallest positive integer m with at least $2d$ digits, satisfying the following conditions:

- $m \geq n$
- m is a concatenation of two prime numbers, each number has at least d digits.

For example, given $d = 2$, $n = 1112$, then the smallest m is 1113 which is the concatenation of two 2-digit primes 11 and 13.

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 consists of one line containing two space-separated positive integers n and d ($1 \leq n \leq 10^9$ và $1 \leq d \leq 5$).

Output

For each data set write in one line the desired number m.

Sample Input	Sample Output
3	**1113**
1112 2	**22**
10 1	**2311**
2011 2	

Problem D: Bumps

In the famous game show name "Bumps", n contestants play together to answer some quiz questions and accumulate points. The best part is that for the last question, the contestants are given a chance to 'bump' up their scores by betting a part of their current points. If they answer the final question correctly, they will be awarded the point. On the other hand, if they answer the question wrongly, the betting amount will be deducted from their current point. The winner is the one whose score is strictly greater than all the other opponents.

Assuming all the opponents have an equal chance of 50% that they would answer the final question correctly. Given everyone's current scores and all the other opponents' amount of bet, your task is to write a program to determine the integer amount of bet that gives you the highest probability of winning**.** If there are multiple solutions that give you the same highest probability of winning, return the smallest such value.

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 200. The following lines describe the data sets.

Each data set consists of three lines, the first line contains the number of players, n, $(2 \le n \le 10)$. On the second line, there are exactly n integers, si $(i=1..n)$, describing the players' current scores $(0 \le si \le 10000)$ where $s1$ is your score. On the third line, there are exactly $n-1$ integers $bi(1 \le i \le n-1)$, describing the amount of bet set by your opponents (bi is the amount of bet of the player whose score is $si+1$, $0 \le bi \le si+1$ for $1 \le i \le n-1$).

Output

For each data set write on one line the amount you want to bet for the final question.

Sample Input	Sample Output
3 2 11 5 5 2 5 11 0 3 100 100 100 25 75	0 0 76

Problem E: Name Split

In the Ancient Country of Mystery (ACM), the rule for naming a newborn baby is to use a string with no duplicate character. For example, *huong* and *hoan* are valid names while *nhung* and *robert* are invalid ones.

Recently, we found a very old document with a lot of names inside. Unfortunately, the names are written continuously without spaces. Perhaps there was no space character in their alphabet. Nevertheless, we would like to help to recover the name list.

Given a string of no more than 50 lowercase characters ('a'-'z') containing the concatenated valid names, your task is to write a program to identify valid names and present them in a pretty format using the following procedure:

- Split that string into smallest number of names conforming to the criteria above.
- To avoid the problem of identifying boundary between names, the first character of each name should be changed to upper case.
- The name list is sorted in alphabetical order.
- This list of sorted names is converted to a string with names concatenated.
- If there are more than one way of splitting into smallest number of names, use the way that generates the smallest string when sorted alphabetically where upper case characters are considered smaller than lower case characters.

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. Each of the following lines describes one data set.

Each data set consists of a single line which contains a string with at most 50 lowercase characters ('a'-'z').

Output

For each data set, write one line the desired name list separated by comma.

Sample Input	Sample Output
4	**AAAAA**
aaaaa	**HuongToan**
toanhuong	**HuyHuyHuyHuyHuy**
huyhuyhuyhuy	**AAb**
aba	

Problem F: Alien Message

Citizens on the ABC planet communicate with each other using a special kind of language. Each message in the ABC language is a sequence of numbers, which are in the range of 1 to n. Recently, people from the Earth have sent a spaceship to visit the ABC planet. They find that it takes so much time to write a message in ABC language. Therefore, they write a software that can help to write a message in ABC language faster. The software allows the user to write a message with two operations:

- Operation 1: Append one number to the end of the current message.
- Operation 2: copy one sub-sequence of numbers within the current message and append it to the end of the message.

For example, the message **1 2 1 2** can be written with three operations:

Operation	**Result**
Append **1** to the end of the current message	**1**
Append **2** to the end of the current message	**1 2**
Copy the sequence **1 2** and append to the end of the message	**1 2 1 2**

Given a message in the ABC language, your task is to write a program to find the minimum number of operations needed to write the message.

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 consists of two lines where the first line contains two space-separated integers n ($n \le 100$) and m ($m \le 10^6$). m represents the message length. The second line contains the sequence of numbers separated by space representing the content of the message.

Output

For each data set write in one line the minimum number of operations required to write the given message.

Sample Input	**Sample Output**
2	**3**
5 4	**6**
1 2 1 2	
2 10	
1 1 1 1 0 0 1 1 0 0	

Problem G: Factorization

Given two positive integer *m* and *k*, a positive integer N of the form $p_1^k \times p_2^k \times \ldots \times p_m^k$, where p_1, p_2, .., p_m are different primes, can be called a "balance number". We call $N = x_1 \times x_2 \times \ldots \times x_l$ a factorization of N if $l > 1$ and $1 < x_1 \le x_2 \le \cdots \le x_l$. Two factorizations $N = x_1 \times x_2 \times \ldots \times x_l$ and $N = y_1 \times y_2 \times \ldots \times y_h$ are different if there exist *i* such that $i \le \min(l, h)$ and $x_i \ne y_i$.

For example, $N = 30 = 2^1 \times 3^1 \times 5^1$ can be factorized in the four different ways:

$30 = 2 \times 3 \times 5$

$30 = 2 \times 15$

$30 = 3 \times 10$

$30 = 5 \times 6$

Your task is to write a program to count the number of different factorizations of a "balance number" 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 100. The following lines describe the data sets.

Each data set consists of only one line, which contains one "balance number" N $(N \le 10^{12})$.

Output

For each data set, write in one line the number of different analyses of the given *N*.

Sample Input	Sample Output
3	**0**
5	**4**
30	**8**
100	

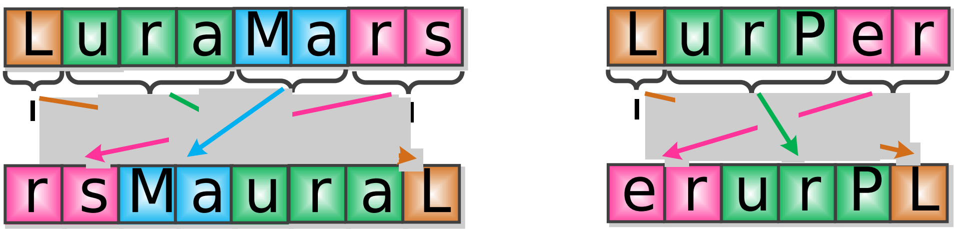
Problem H: Decoding

The two communities A and B are trying to expand and establish its sovereignty to other planets in the universe. Community A has an ordered list of planets to attack called expansion plan. In order to send the expansion plan to commanders electronically, they represent the expansion plan as string by concatenating all planets' names in order where each planet's name begins with a single uppercase letter followed by lowercase letters. More formally, if the planets' names are $p_1,p_2,...,p_n$ then the expansion plan is the string $p_1p_2...p_n$

To keep their expansion plan secret from community B, they encode the string by swapping character groups using the following scheme:

1. Initially, the string is divided into several groups. The 1st group consists of the 1st character; the 2nd group consists of the last 2 characters. Subsequently, the ith group consists of exactly *i* characters immediately after the *(i-2)th* group if *i* is odd or before the *(i-2)th* group if *i* is even. The process is repeated until all characters in the original string have been allocated to one group. Note that there might not be enough characters for the last group according to the above rule; in that case, the last group will contain all the remaining characters. The groups must not overlap each other i.e. every character belongs to one and only one group.
2. Next we swap groups 1 and 2, group 3 and 4, group 5 and 6, etc. If the number of groups is odd, the last group remains unchanged.

The figures below demonstrate the encoding scheme. Note that this encoding does not change the length of the string.



Community B wants to compete with community A by finding out the expansion plan of A and attack those planets before community A executes their plan. Community B was able to intercept the encoded expansion plan of A as well as the encoding scheme to identify A's expansion plan. However, due to limited resources, they were not able to attack all planets in A's expansion plan. Instead they could only attack planets having prime indices in the expansion plan of A. More formally, if A's expansion plan (not encoded) is a name sequence $p_1p_2...p_n$, then the expansion plan of B is a name sequence $p_{i1}p_{i2}...p_{im}$ where i_k are all prime numbers between 2 and n inclusively i.e. $1 < i_k \le n$, $i_k < i_{k+1}$, $k = 1 \div m$.

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 25. The following lines describe the data sets.

Each data set consists of one line containing the string of no more than 5×10^5 characters representing the encoded plan of A.

Output

For each data set, write in one line the string representing the expansion plan of B. If B's plan is empty then you should write "***Impossible***" instead.

Sample Input	Sample Output
3	**Mars**
rsMauraL	**Per**
erurPL	**MarsCassiopeia**
iaiopeassMarsCunaL	

Problem I: Viruses

Professor John Disket's group is conducting experiments about emerging diseases caused by combinations of different virus types. To this end, they sets up virus of N different types $V_1,\ldots V_n$ at N points $(x_1, y_1)\ldots(x_n,y_n)$ on the plane, respectively. After an hour, virus of each type V_i spreads and occupies a circle (including the perimeter) with radius r_i and center at (x_i, y_i).

Your task is to write a program to help Professor John Disket's group to find out if there exists any point occupied by all these N viruses after one hour.

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 a positive integer N ($N \leq 100$). The i^{th} line of the following N lines contains 3 integers x_i, y_i, r_i ($-10^9 < x_i, y_i < 10^9$, $0 < r_i < 10^9$) separated by a space.

Output

For each data set, write in one line a number 1 if there exists a point occupied by all these viruses. Otherwise, write in one line a number 0.

Sample Input	Sample Output
2 **2** **0 0 1** **0 2 1** **3** **0 0 1** **0 2 1** **5 5 1**	**1** **0**

Problem J: Page Layout

A page can be represented as a matrix of W columns which is numbered 1, 2,.. W from left to right, and unlimited number of rows which is number 1,2,… from top to bottom. Each cell of the matrix can contain one character. We have to place on the page an article containing N words, each of which consists of L_i characters. Each i[th] word is placed in the first position (from top to bottom, then from left to right) that satisfies the following rules:

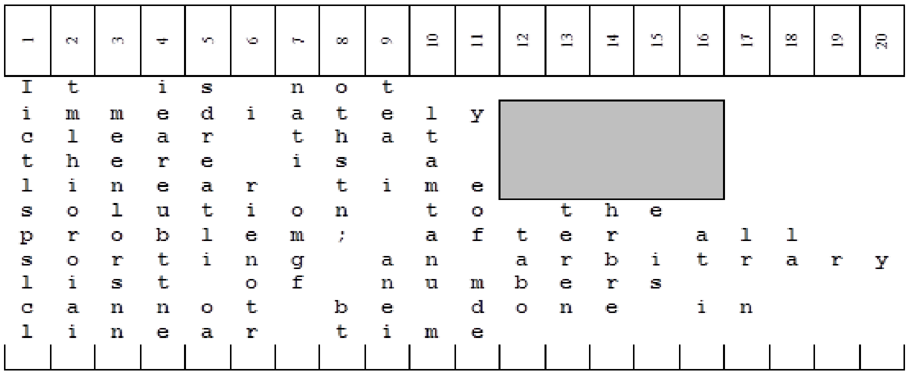
- Each cell can contain only one character of one single word;
- The characters of the i[th] word must be placed on L_i consecutive cells in a row of the page;
- The last character of the i[th] word and the first character of the (i+1)[th] word can't be in adjacent cells of the same row ($1 \le i < N$);
- The i[th] must be placed before the (i+1)[th] word ($1 \le i < N$), which means the index of the row containing the characters of the i[th] word must be smaller than that of the row containing the characters of (i+1)[th] word, or the index of the rows are be same and the index of the column containing the first character of the i[th] word must be smaller than that of the column containing the first characters of (i+1)[th] word;

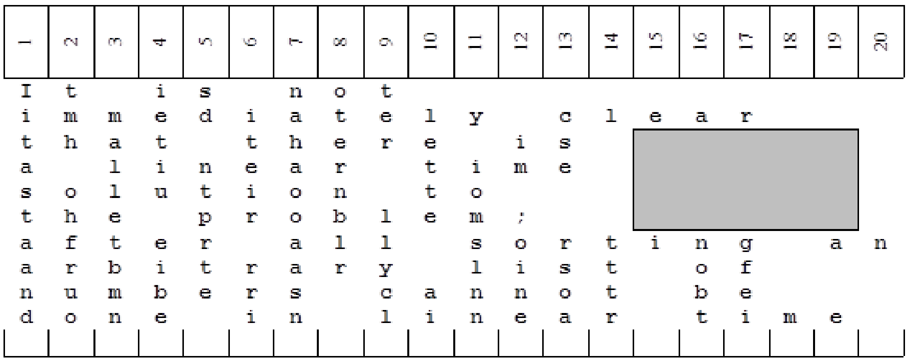
For example, a sample article can be placed on a page with 20 columns as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I	t		i	s		n	o	t											
i	m	m	e	d	i	a	t	e	l	y		c	l	e	a	r			
t	h	a	t		t	h	e	r	e		i	s		a					
l	i	n	e	a	r		t	i	m	e		s	o	l	u	t	i	o	n
t	o		t	h	e		p	r	o	b	l	e	m	;					
a	f	t	e	r		a	l	l		s	o	r	t	i	n	g		a	n
a	r	b	i	t	r	a	r	y		l	i	s	t		o	f			
n	u	m	b	e	r	s		c	a	n	n	o	t		b	e			
d	o	n	e		i	n		l	i	n	e	a	r		t	i	m	e	

In addition to the article, a picture has to be placed on the page. The picture takes up a rectangular area of R rows high and C columns wide. The picture is placed on the page before the text is laid out and it may be placed anywhere on the page. After the picture is placed, the characters of the article cannot be placed on the cells occupied by the picture.

A row on the page is considered used if, after the picture is placed and the article is laid out, at least one cell of the row is occupied. The total number of rows used may depend on the position of the picture. For example, two possible final layouts of the picture and the text are shown below, with one using 11 and the other 10 rows:





Given information about the article and dimentions of the picture, your task is to write a program to determine the minimum number of row needed to lay out the article and the picture.

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 starts with the first line containing four integers N, W, R, C ($1 \le N \le 10000$, $1 \le W, R \le 1000$, $1 \le C \le W$) separated by space, followed by several lines containing N integers l_1, l_2, ... l_N ($1 \le l_i \le W$) where l_i represents the character length of the i^{th} word of the article. The numbers may be separated by spaces and/or newlines.

Output

For each test case, output a single line containing the minimum number of rows needed to layout the article.

Sample Input	Sample Output
1 **26 20 3 6** **10 7 2 1 4 6 4 4 3 5 3 3** **7 6 7 4 2 7 4 10 3 6 5 8 7 7**	**10**