Problem A. Allo

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given a string, consisting of digits from 0 to 9. Find out all digits, which are not used in this string.

Input

First line of the input file contains one integer T — number of the test cases ($1 \leq T \leq 150$). Each of next T lines contains one test case — non-empty string, consisting of digits from 0 to 9. Length of the given string does not exceed 30.

Output

For each test case print in accending order all digits, which did not used in this string. If all ten digits are used, print "`allo`" instead.

Examples

standard input	standard output
2 2468 0123456789	013579 allo

Problem B. Binary Multiplication

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Implement algorithm of the long multiplication (one was used to multiplicate on the paper long time before computers were invented) for the binary integers.

Input

First line of the input contains one integer T — number of test cases ($1 \le T \le 150$). Each of next T lines contain two integers in the binary representation without leading zeroes, length of each does not exceed 30 digits.

Output

For each test case, print line with text "`Test N`", where N is number of test. Then print log of the multiplication according to the samlple. Answers to the sequential tests must be separated by the extra blank line.

Examples

standard input

```
4
11 11
111 10
10 111
1010 0
```

standard output

```
Test 1
  11
  11
  --
  11
 11
----
1001

Test 2
 111
  10
 ---
 000
111
----
1110

Test 3
  10
 111
 ---
  10
 10
10
----
1110

Test 4
1010
   0
----
0000
----
   0
```

Problem C. Code

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

In this problem transpositions are represented as sequence of N pairwise distinct positive integers, not greater than N.

Defile $L(P)$ for an transposition P of the N elements as sequence of N integers, such as $L_i(P) = |j : P_j < P_i,\ j > i|$. In other words, i-th element of $L(P)$ denotes number of elements of P right from i-th element in the representation of P, which are strictly lesser than i-th element of P.

Given P, calculate $L(P)$.

Input

First line of the input contains number of test cases T ($1 \le T \le 20$). Each test cases consists of two lines. First one contains a transposition of length N ($1 \le N \le 10^5$). It is guaranteed that sum of all N in the input file does not exceed 10^6.

Output

For each testcase print in separate line N numbers — values of $L(P)$ for the given transposition.

Examples

standard input	standard output
1 5 5 4 3 2 1	4 3 2 1 0

Problem D. Driving

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Road system of the Byteland consists of N crossings and M bidirectional roads. Each road connects some crossing with another one, and there are no more than one direct road between two crossings. It is guaranteed that in the road system exists atleast one cycle (i.e. sequence of roads (c_1, c_2), (c_2, c_3), $\ldots$, (c_n, c_1), where (a, b) denote direct road connecting crossings a and b, $n > 2$ and all c_i are pairwise distinct).

Find out the shortest cycle in the Byteland road system, i.e. cycle with mininal n.

Input

First line of the input contains one integer T — number of test cases ($1 \le T \le 20$). First line of each test case holds two integers N and M ($1 \le N \le 200$, $1 \le M \le 1000$). Each of next M lines defines a road and contains two integers A and B — numbers of crossings, connected by this road ($1 \le A, B \le N$, $A \ne B$). It is guaranteed that sum of all M in the input file does not exceed 7500.

Output

For each test case print one integer — minimal number of the roads in the cycle.

Example

standard input	standard output
1 3 3 1 2 2 3 1 3	3

Problem E. The Great Equalizer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

The Great Equalizer is tool for the transformation of integers. In one step the Great Equalizer can do one of following operations:

- Add one of the given integers $a_1, \ldots a_k$.
- Multiply by one of the given integers $b_1, \ldots, b_s$.

Find out number of distinct ways to use the Great Equalizer to obtain integer M from the integer N, if as last operation was used multiplication. Note that two ways differ if they contain different number of steps or if at some step was used different operation or different integer from appropriate set.

Input

First line of the input file contain integer T — number of the test cases ($1 \leq T \leq 70$). Next T test cases follow. First line of each test case contains two integers N and M ($1 \leq N < M \leq 10^5$). Next line contains one integer K ($0 \leq K \leq 100$). If $K > 0$ then next line contains K positive integers $a_1, \ldots, a_k$ ($1 \leq a_i \leq 100$). Next line contain integer S ($0 \leq S \leq 100$). If $S > 0$, then next line contains S positive integers $b_1, \ldots, b_k$ ($2 \leq b_i \leq 100$). It is guaranteed that sum of all $(K+S)$ for all test cases is not greater than 500, sum of M for all tests is not greater than $1.3 \cdot 10^6$.

Output

For each test case print in separate line one integer — number of ways to transform N to M, if the multiplication was used at last step. Print answer modulo 10^9 (because jury of East Siberian Subregionals does not know about primes).

Example

standard input	standard output
3 1 5 3 1 2 3 2 4 5 1 8 0 2 2 4 1 6 2 1 1 2 3 3	1 3 1

Problem F. Factory

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

N machines are installed on the factory. Each machine produces identical details. For each machine is given time in the minutes to produce one detail. Machines are working independently from each other. Find out mininal time in minutes, needed for production of K details; you may use one machine as well as several simultaneously working ones.

Input

First line of the input contains one integer T — number of the test cases ($1 \le T \le 700$). First line of the test case contains two integers: N — number of machines ($1 \le N \le 10^5$) and K — number of details to produce ($1 \le K \le 10^9$). Second line contains N positive integers t_i — time in minutes per one detail for i'th machine. All t_i does not exceed 10^9, sum of all N's in the input file does not exceed $2 \cdot 10^9$.

Output

For each test case print in the separate line one integer — minimal time in minutes, needed to produce K details.

Example

standard input	standard output
1 2 1 2 2	2

Problem G. Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Anonymous and Bredor are playing in the next game. Initially, number 1 is given. On his turn each player must multiply current number by one of integers between 2 and 9, inclusively. Goal is to obtain number not less than given integer N. Player, who obtained such a number first, is declared as winner. Anonymous always starts first.

Find out, who will win if Anonymous and Bredor will play optimally.

Input

First line of the input contains one integer T — number of the test cases ($1 \le T \le 2500$). Each of next T lines contain one integer N ($2 \le N \le 10^9$).

Output

For each test case print in separate line 1, if Anonymous will win the game, and 2 otherwise.

Example

standard input	standard output
4 9 10 1149729 999999999	1 2 2 1

Problem H. Holidays

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

During long school holidays a smart young boy Vasya plays the following game. Initially N piles of stones are given, each of them containing a_i stones. In one turn Vasya can do exactly one of the following actions:

- Remove from 1 to 9 stones from any single pile (of course, if there are enough stones there).
- If the number of stones in each individual pile a_i is a factor of 10, then remove 9/10 stones from each pile. Or more formally, if $a_i = 0(\mod 10)$ for all a_i, replace all a_i by $a'i = a_i/10$, where a_i is the number of stones in the pile i.

Goal of the game is to empty all piles (i.e. make all a_i equal to zero) using minimal number of turns. Please help Vasya to calculate this number.

Input

First line of the input contains positive integer T — the number of test cases ($1 \le T \le 210$). Then T test cases follow. Each test case contains one or two lines. First line contains integer N — number of piles ($0 \le N \le 100$). If $N > 0$ then second line contains N space-separated integers a_i — amount of stones in an individual pile ($0 \le a_i \le 10_9$), otherwise, there is no second line.

Output

For every test case on a separate line print the minimum number of turns that Vasya has to make in order to empty all piles, assuming he plays optimally.

Examples

standard input	standard output
3 2 1 6 3 10 4 1 3 15 201 45	2 4 8

Problem I. Interesting Integers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Undoubtedly you know of the Fibonacci numbers. Starting with $F_1 = 1$ and $F_2 = 1$, every next number is the sum of the two previous ones. This results in the sequence $1, 1, 2, 3, 5, 8, 13, \ldots$

Now let us consider more generally sequences that obey the same recursion relation

$$G_i = G_{i-1} + G_{i-2} \; for \; i > 2$$

but start with two numbers $G_1 \leq G_2$ of our own choice. We shall call these Gabonacci sequences. For example, if one uses $G_1 = 1$ and $G_2 = 3$, one gets what are known as the Lucas numbers: $1, 3, 4, 7, 11, 18, 29, \ldots$. These numbers are — apart from 1 and 3 — different from the Fibonacci numbers.

By choosing the first two numbers appropriately, you can get any number you like to appear in the Gabonacci sequence. For example, the number n appears in the sequence that starts with 1 and $n - 1$, but that is a bit lame. It would be more fun to start with numbers that are as small as possible, would you not agree?

Input

On the first line one positive number: the number of test cases, at most 100. After that per test case one line with a single integer n ($2 \leq n \leq 10^9$) follows: the number to appear in the sequence.

Output

For each test case print one line with two integers a and b ($0 < a \leq b$), such that, for $G_1 = a$ and $G_2 = b$, $G_k = n$ for some k. These numbers should be the smallest possible, i.e., there should be no numbers a' and b' with the same property, for which $b' < b$, or for which $b' = b$ and $a' < a$.

Example

standard input	standard output
5 89 123 1000 1573655 842831057	1 1 1 3 2 10 985 1971 2 7

Problem J. Job

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Igor is a young promising programmer who was employed in city's leading software development company ACM (Agency of Creative Masters). Suddenly, the company has just moved into new large office. Office room has rectangular form with dimensions $N \times M$ meters. There are multiple rectangular bookcases in the office placed parallel to office's walls.

After company has moved, many new working tables were bought. Each working table can be seen on the office plan as a rectangle 2×3 meters. The tables are to be placed in the office given the following restrictions:

- Tables' sides are parallel to office's walls.
- Tables' corners are located in points with integer coordinates.

There must be at least one meter of free space between any point of table and wall, another table, or bookcase. Bookcases must keep their initial positions.

The boss wants the employees to place as many working tables in the office as possible given the restrictions above. You are to find this number.

On the figure below, one of possible correct arrangements is shown, there are two tables and two bookcases (this is one of possible solutions for example case).

```
BBB.....
.....TT.
.....TT.
.TTT.TT.
.TTT....
.....BB.
```

Input

First line of the input contains integer T — number of test cases to compute. Then T tests follow ($1 \leq T < 150$).

First line of the test case contains three integere N, M and B, where N and M are dimensions of the office room ($1 \leq N, M \leq 10$), B is the number of bookcases ($0 \leq B \leq 10$). Then B lines follow, each describing a bookcase. Bookcase description contains integers X, Y, W, H, where X and Y are zero-based positions of a bookcase in the room ($0 \leq X < N$, $0 \leq Y < M$), and $W \times H$ are dimensions of a bookcase ($1 \leq W \leq N$, $1 \leq H \leq M$).

Bookcases can touch but not intersect each other and walls.

Output

For each test case print one integer T — maximal number of tables that can be placed in the room obeying the restrictions above.

Example

standard input	standard output
1 6 8 2 0 0 1 3 5 5 1 2	2

Problem K. Koh-I-Noor

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given N pairwise different Koh-I-Noor pencils and K boxes. Each box have enough capacity to hold all N pencils. Find number of ways to put pencils in boxes, if boxes cannot be distinguished one from another and no box bust remain empty.

Input

First line of the input contains one integer T — number of tests ($1 \le T \le 500$). Each of next N lines contain two integers: N — number of pencils ($1 \le N \le 10^9$) and K — number os boxes ($1 \le K \le min(N, 10)$).

Output

For each test case print in the new line one integer — number of ways to distribute pencils by boxes modulo 2017.

Example

standard input	standard output
2 1 1 3 2	1 3

For second sample three ways are (1)(2,3), (2)(1,3) and (3)(1,2).