Problem A
Rush Hour Puzzle

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Rush Hour is a puzzle game invented by Nob Yoshigahara in the 1970s. It is now being manufactured by **ThinkFun**. The board is a 6×6 grid with grooves in the tiles to allow vehicles to slide. Cars and trucks are both one square wide, but cars are two squares long and trucks are three squares long. Vehicles can only be moved forward or backward along a straight line on the grid. The goal of the game is to get the only red car totally out through the exit of the board by moving the other vehicles out of its way. Figure 1 gives an example of Rush Hour puzzle.

Figure 1: An example of Rush Hour puzzle.

We give each vehicle of a puzzle a unique *id*, numbered from 1 to the number of vehicles, in which the red car's id is 1. The board information of a puzzle is represented by a 6×6 matrix, named *board matrix*. Each entry of a board matrix is the id of the vehicle placed on that groove, and the entries are filled with 0 if there exists no vehicle on those grooves. The exit of the board is located at the right end side of the 3rd row. Figure 2 shows the board matrix corresponding to the puzzle in Figure 1.

Moving a piece (car or truck) by one unit (a groove) is called a *step*. A puzzle is *easy* if it can be solved (the red car totally out through the exit of the board) in no more than 10 steps. Please write a program to judge whether a puzzle is easy or not.

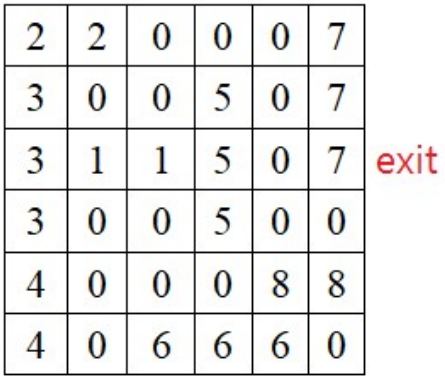

Figure 2: The board matrix corresponding to the puzzle in Figure 1.

Input Format

The input contains 6 lines, each line indicates the content (6 integers separated by a blank) of each row of a board matrix.

Output Format

Output the minimum number of steps for solving the input puzzle if the puzzle is easy, otherwise output `-1`.

Technical Specification

- There are at most 10 vehicles on the board for each puzzle.
- Only the red car can be moved out of the board for each puzzle.

Sample Input 1

```
2 2 0 0 0 7
3 0 0 5 0 7
3 1 1 5 0 7
3 0 0 5 0 0
4 0 0 0 8 8
4 0 6 6 6 0
```

Sample Output 1

```
-1
```

Sample Input 2

```
0 2 0 6 6 0
0 2 0 0 7 0
0 3 1 1 7 0
0 3 4 4 8 0
0 5 5 5 8 0
0 0 0 0 0 0
```

Sample Output 2

```
6
```

Problem B
The Power Monitor System

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

An electric power system contains *electrical nodes* and *transmission lines*. An electrical node is a substation bus where transmission lines, loads, and generators are connected, and a transmission line joins two electrical nodes. To ensure electric power systems working normally, electric power companies need to continually monitor their system's state by a set of state variables such as the voltage magnitude at loads and the machine phase angle at generators. Therefore, they place PMUs at selected electric nodes in the system for monitoring these variables (voltage and phase angle).

A *phase measurement units* (PMU) measures the state variables for the electric node at which it is placed and its incident transmission lines and their end-electric-nodes. These nodes and lines are said to be *monitored*. Other nodes and lines are monitored recursively by the following rules:

1. Any node that is incident to a monitored line is monitored.
2. Any line joining two monitored nodes is monitored.
3. If a node is incident to a total k lines, where $k > 1$, and $k-1$ of these lines are monitored, then all these k lines are monitored.

Because of the high cost of a PMU, it is desirable to minimize their number while maintaining the ability to monitor the entire system. For example, the placement of PMUs (black points) satisfies the requirement that the entire system is monitored in Figure 3(a) but not in Figure 3(b). In Figure 3(a), nodes 1, 2, 3, 4 and lines a, b, c are monitored by PMU1, nodes 6, 8, 9, 10 and lines g, h, i are monitored by PMU2. According to Rule 2, line e is monitored since nodes 4 and 6 are monitored. According to Rule 3, lines d and f are monitored because they are the last monitored lines incident to node 4 and node 6, respectively. Nodes 5 and 7 are monitored according to Rule 1.

In Rich City, the electric power system forms a tree structure (a structure without loops). Please write a program to compute the minimum number of PMUs to monitor the entire system.

Input Format

The first line of the input is an integer n, $2 \leq n \leq 100000$, which indicates the number of electrical nodes. Each of the next $n-1$ lines consists of two integers separated by a space, which indicates the indices of two adjacent electrical nodes of a transmission line.

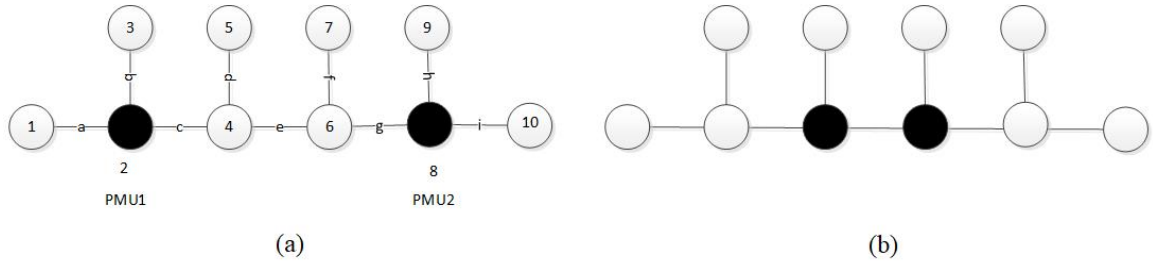

Figure 3: (a) Satisfied placement. (b) Unsatisfied placement.

Output Format

Output the minimum number of PMUs, which satisfies the requirements of the problem.

Technical Specification

- There are n nodes, $2 \leq n \leq 100000$.
- Each node has a unique index between 1 and n, which are labeled from a root node in breadth-first traversal order.

Sample Input 1

```
10
1 2
2 3
2 4
4 5
4 6
6 7
6 8
8 9
8 10
```

Sample Output 1

```
2
```

Sample Input 2

```
5
1 2
1 3
1 4
1 5
```

Sample Output 2

```
1
```

Problem C
Are They All Integers?

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Computing using integers is a dream for every programmer. That is, you do not have to deal with floating point numbers, estimated errors, and etc. We do not even need any floating point units in our computers for divisions!

Your company claimed there is a brand new computational model that solves integer problems efficiently. As a software engineer in this Integer Computing Processors Company (ICPC), you are going to write a validator that checks the following:

Given a list of positive integers $A[0], \ldots, A[n-1]$. Suppose you pick three different elements arbitrarily of this list, $A[i], A[j], A[k]$ with i, j, k being mutually different. Is it true that $\frac{A[i]-A[j]}{A[k]}$ is always an integer?

Input Format

The first line of the input contains one integer n indicating the number of positive integers in the list. The second line of the input contains n positive integers $A[0], \ldots, A[n-1]$ separated by blanks.

Output Format

If $\frac{A[i]-A[j]}{A[k]}$ is always an integer, then output `yes`. Otherwise output `no`.

Technical Specification

- $3 \leq n \leq 50$
- $1 \leq A[0] \leq A[1] \leq \cdots \leq A[n-1] \leq 100$

Sample Input 1

```
5
1 1 1 1 4
```

Sample Output 1

```
yes
```

Sample Input 2

```
5
1 2 4 8 16
```

Sample Output 2

```
no
```

Almost blank page

Problem D
Tapioka

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Nowadays, bubble tea has been famous around the globe. They have different names – bubbles, pearls, and tapioka. People start thinking to put almost every food together with tapioka. E.g., bubble tea pizza, bubble tea ramen, bubble tea hotpot, bubble tea cake, and so on.

Given the name of a dish, you are going to *repeatedly* remove *all* the tapioka part and reveal the true colors of that dish. Notice that the dishes in this problem will consist of *exactly* three words. Every dish always starts with either the term "bubble tea" or "tapioka". For example, you may observe a dish named "bubble tea ramen", but you will never observe a dish named "bubble ramen tea".

Input Format

The input file contains exactly one test case. There is only one line in the input file, and it contains a dish name consisting of exactly three words.

Output Format

If after repeatedly removing all occurrence of "bubble" or "tapioka" (as entire word) but you found nothing left, output "nothing". Otherwise, output the remaining words separated by blanks in the original order.

Technical Specification

- In this problem, a word is a string of English letters in lowercase.
- In the input file, all words are seperated by blanks.
- The length of each word does not exceed 32.

Sample Input 1

```
bubble tea pizza
```

Sample Output 1

```
tea pizza
```

Sample Input 2

```
tapioka cake tapiokas
```

Sample Output 2

```
cake tapiokas
```

Sample Input 3

```
tapioka jasmine tea
```

Sample Output 3

```
jasmine tea
```

Sample Input 4

```
tapioka bubble tapioka
```

Sample Output 4

```
nothing
```

Almost blank page

Problem E
The League of Sequence Designers

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Consider the following sequence problem: given n integers $a_1, a_2, \ldots, a_n$, where $|a_i| \leq 10^6$ for all $1 \leq i \leq n$ and $1 \leq n < 2000$, compute

$$\max_{1 \leq \ell \leq r \leq n} (r - \ell + 1) \cdot \sum_{\ell \leq i \leq r} a_i.$$

As an attempt to solve the above problem, Natasha came up with a textbook greedy algorithm using the idea of computing heaviest segment via prefix sums as follows:

Function `FindAnswer`$(a_1, a_2, \ldots, a_n)$

```
result = 0
curMax = 0
left = 0
for i = 1 to n do
    curMax = curMax + a_i
    if curMax < 0 then
        curMax = 0
        left = i
    end
    result = max( result, (i − left)× curMax )
end
return result
```

As you can see, Natasha's idea is not entirely correct. For example, when the input sequence is $6, -8, 7, -42$, the function `FindAnswer` will return 7, but the correct answer is $3 \cdot (6-8+7) = 15$.

Bruce tries to tell Natasha that her solution is not correct, but she does not believe.

Given an integer k and a lower bound of sequence length L, your task in this problem is to help Bruce design a sequence of n integers with $n \geq L$ such that the correct answer and the answer produced by Natasha's algorithm differ by exactly k.

Note that, the sequence you produce must follow the specification to the original problem. That is, $1 \leq n < 2000$ and $|a_i| \leq 10^6$ for all $1 \leq i \leq n$. Print `-1` if it is impossible to form such a sequence.

Input Format

The input file starts with an integer T, the number of testcases, in the first line.

Then there are T lines, one for each testcase, each containing two integers k and L, separated by a space.

Output Format

The output for each testcase consists of either one or two lines, depending on the result. The format is as follows.

If there exists no such sequences, print the integer `-1` in a line. Otherwise, print in the first line the integer n denoting the length of the sequence. In the second line, print the n integers $a_1, \ldots, a_n$ separated with a space.

Technical Specification

- $1 \leq T \leq 5$
- $1 \leq k \leq 10^9$
- $0 \leq L \leq 10^9$

Sample Input 1

```
3
8 3
612 7
4 2019
```

Sample Output 1

```
4
6 -8 7 -42
7
30 -12 -99 123 -2 245 -300
-1
```

Problem F
Miss Sloane

Time limit: 5 seconds
Memory limit: 1024 megabytes

Problem Description

Elizabeth Sloane, the great cutthroat lobbyist, and the firm she's working in was approached in the movie by the gun manufacturing representative Bill Sanford who leads the opposition to the proposed Heaton-Harris bill that would expand background checks on gun purchases. Sanford proposes specifically to target female voters in order to overturn their Maternal tendency when voting. Miss Sloane ridiculed this idea due to her belief that the power of congressmen should be exercised and influence in a manner that benefits everyone, not just the congressmen themselves, and that is the right thing to do.

In this problem you are to play the other scenario for which Miss Sloane takes Bill Sanford's proposal to campaign against the Heaton-Harris bill. Your goal is to lobby the senators in the congress in a way that prevents them from reaching an agreement on the bill! There are n senators in the congress, each of which possesses an *indicator* a_i, which is a positive integer, for his/her general opinions towards various public issues. The *resistance* of each senator against lobbying is indicated by another positive integer e_i. When the greatest common divisor of the public-issue indicators is not trivial, i.e., not 1, an agreement among the senators will be reached and the vote for Heaton-Harris bill will pass. Apparently your goal is to prevent this from happening.

Miss Sloane's ability to lobby is indicated by an integer k. The campaign goes as follows: In each round, Miss Sloane has the option to approach one senator, say, the i^{th} senator, and influence his/her opinions towards certain public issues of Sloane's choice. As a result, Miss Sloane can divide the public-issue indicator of the i^{th} senator, a_i, by a natural divisor that is no larger than Sloane's ability to lobby. The senators, however, are no fools. Hence, Miss Sloane can influence (be it by trapping/blackmailing) each of them at most once. Furthermore, as the campaign proceeds and more senators are lobbied, it will take substantially more time to further trap (influence) other senators. If Miss Sloane has already lobbied x senators that have a total resistance y, and she wants to further lobby the i^{th} senator, then the lobbying for the i^{th} senator will take $y + e_i \cdot (x + 1)$ units of time to complete.

Please compute for Miss Sloane the minimum time to accomplish the campaign or print `-1` if this is impossible.

Input Format

The first line consists of two integers n and k, the number of senators and Miss Sloane's ability to influence. The second line contains n integers $a_1, a_2, \ldots, a_n$ which are the indicators of the n senators. The third line contains n integers $e_1, e_2, \ldots, e_n$ which are the resistances of the senators.

Output Format

Print the minimum units of time to accomplish the campaign, or `-1` if this is impossible.

Technical Specification

- $1 \leq n \leq 10^6$
- $1 \leq k \leq 10^{12}$
- $1 \leq a_i \leq 10^{12}$
- $1 \leq e_i \leq 10^9$

Sample Input 1

```
3 6
30 30 30
100 4 5
```

Sample Output 1

```
18
```

Sample Input 2

```
1 1000000
1
100
```

Sample Output 2

```
0
```

Sample Input 3

```
3 5
7 7 7
1 1 1
```

Sample Output 3

```
-1
```

Problem G
Optimal Selection

Time limit: 8 seconds

Memory limit: 1024 megabytes

Problem Description

Given n distinct numbers, one can output the k-th smallest number without sorting. This problem is known as *selection problem*, and can be solved in linear time. Some people argue that the existing algorithms for selection is too slow to put into practice. So the conjecture is, every time you encounter an algorithm that uses selection as a building block, you can feel free to replace selection with a fast sorting procedure. However, this conjecture seems not to be true.

Bob is trying to answer whether the above conjecture is valid, but it turns out to be a complicated problem that requires many trial and error experiments. After some failed trials, he decides to first focus on finding the best algorithm to implement selection. That is to find an algorithm that implements selection with the least number of comparisons. We assume that the n input numbers will be all distinct. For $n = 3$ and $k = 1$, an optimal algorithm to solve the selection problem is shown as follows:

```
double selection(double a[3], int k = 1){
    if(a[0] < a[1]){
        if(a[0] < a[2]){
            return a[0];
        }else{
            return a[2];
        }
    }else{
        if(a[1] < a[2]){
            return a[1];
        }else{
            return a[2];
        }
    }
}
```

For every possible $(a[0], a[1], a[2])$, the above selection function uses at most 2 comparisons. Indeed, 2 comparisons is the best possible if the relative order between two input numbers can be inferred only by comparisons, aka *comparison-based algorithms*. So the *complexity* of the selection function for $(n, k) = (3, 1)$ is 2. Now, Bob is considering a more general problem that, the comparison results between some pairs from the input are known before invoking the selection function. The *complexity* of the selection function is defined as the worst case number

of comparisons required, by an optimal comparison-based algorithm, on any possible consistent input array. That is, Bob is wondering what is the complexity of the selection function, given n, k, and some relative order among the n input numbers.

Input Format

Three integers n, k, and ℓ are given in the first line. Then ℓ lines follow. Each of the subsequent ℓ lines contains two integers x and y in $[0, n-1]$ specifying that the relative order between $a[x]$ and $a[y]$ is known to be $a[x] < a[y]$ before invoking the selection function.

Output Format

Given n, k, and the relative order between the given ℓ pairs, output the complexity of the selection function.

Technical Specification

- $1 \leq n \leq 8$.
- $1 \leq k \leq n$.
- $0 \leq \ell \leq n$.
- The relative order between the given ℓ pairs will be consistent.

Sample Input 1

```
3 2 0
```

Sample Output 1

```
3
```

Sample Input 2

```
7 2 5
0 6
3 6
4 6
2 0
0 5
```

Sample Output 2

```
5
```

Problem H
Mining a

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Blockchain technology is used in many digital currency systems, such as Bitcoin and Ethereum. In this technology, distributed users share a common list of records (called the chain), and a user has the right to add a new record to the chain by solving a mathematical problem. This process is called *mining*. The i-Taiwan company has developed up a new digital currency system, called the ICPC (I-taiwan Coins for the Public Currency). In the ICPC system, its mathematical problem for mining is as follows. Given positive integer n, the problem asks one to find the largest integer a such that

$$\frac{1}{n} = \frac{1}{a \oplus b} + \frac{1}{b}, \text{ for some positive integer } b$$

where $\oplus$ is the bitwise exclusive-or operator. For example, for $n = 12$, its solution is $a = 145$. In this case, $b = 13$ and thus $a \oplus b = 145 \oplus 13 = 10010001_2 \oplus 1101_2 = 10011100_2 = 156$. Accordingly,

$$\frac{1}{a \oplus b} + \frac{1}{b} = \frac{1}{156} + \frac{1}{13} = \frac{1}{12} = \frac{1}{n}.$$

You are an ambitious programmer, and you want to mine a lot of digital coins from this system in a short time. Please write a program to find the largest a for each n in order to earn the rewards from ICPC.

Input Format

The first line of the input file contains exactly one positive integer T that gives you the number of test cases. In the following T lines, each line corresponds to one test case and specifies the integer n.

Output Format

For each test case, output the largest number a in one line.

Technical Specification

- $1 \leq T \leq 20$
- $0 < n \leq 10^7$

Sample Input 1

```
3
6
7
10
```

Sample Output 1

```
45
48
101
```

Sample Input 2

```
3
1
2
7777777
```

Sample Output 2

```
0
5
60493819864864
```

Problem I
The Spectrum

Time limit: 5 seconds

Memory limit: 1024 megabytes

Problem Description

Let $X = (x_1, x_2, \ldots, x_n)$ be an integer sequence whose elements are distinct. The *spectrum* of X, denoted by $spec(X)$, is the multiset $\{|x_i - x_j| : 1 \le i < j \le n\}$. Notice that a multiset counts multiplicity but ignores order. For example, $\{1, 1, 2\}$ and $\{2, 1, 1\}$ are the same, but $\{1, 1, 2\}$ and $\{1, 2\}$ are different in multisets. For simplicity, we assume that sequence X is in the *ascending* order and $x_1 = 0$. For example, suppose $X = (0, 1, 4, 5)$. Then $spec(X) = \{1, 1, 3, 4, 4, 5\}$. Given X, it is easy to compute $spec(X)$. However, given $spec(X)$, it is not an easy task to recover X from $spec(X)$. In fact, it is possible that $spec(X) = spec(Y)$ for two different sequences X and Y. For example, $spec(0, 7, 20) = \{7, 13, 20\} = spec(0, 13, 20)$. Your job is to recover all possible X's such that $spec(X)$ is equal to the specified spectrum in the input.

Input Format

The first line in a test case gives you the number n, which is the size of the integer sequence X. The second line gives you the spectrum of X, which is a multiset and the numbers $d_1, \ldots, d_{\frac{n(n-1)}{2}}$ are listed in nondescending order with a single space as the delimiter between two consecutive numbers.

Output Format

First, output the total number of possible X's (i.e. the number of solutions) in a line. Then dump all possible X's in the lexicographic order (i.e. the dictionary order), one X per line. Let $Y = (y_1, \ldots, y_n)$ and $Z = (z_1, \ldots, z_n)$ be two such solutions. Then Y *should precede* Z if and only if there exists some index k where $1 \le k \le n$ such that $y_k < z_k$ and $y_j = z_j$ for all $1 \le j < k$. For example, the sequence $Y = (0, 7, 20)$ should precede $Z = (0, 13, 20)$ in the lexicographic order because $y_2 < z_2$ (i.e. $7 < 13$) and $y_1 = z_1$. For each X, print its elements in ascending order with a single space between two consecutive numbers.

Technical Specification

- $2 \le n \le 62$.
- $0 < d_1 \le d_2 \le \cdots \le d_{\frac{n(n-1)}{2}}$.
- Your output should satisfy: $0 \le x_i \le 999$ for $1 \le i \le n$ and $x_1 = 0$.
- $x_i < x_j$ for $1 \le i < j \le n$.

Sample Input 1

```
4
2 2 2 4 4 6
```

Sample Output 1

```
1
0 2 4 6
```

Sample Input 2

```
5
3 3 6 9 9 12 12 15 18 21
```

Sample Output 2

```
2
0 3 12 15 21
0 6 9 18 21
```

Sample Input 3

```
4
5 6 7 8 9 10
```

Sample Output 3

```
0
```

Problem J
Automatic Control Machine

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

The company has produced an Automatic Control Machine (ACM for short) that is very popular. Due to its complete and powerful features, the company is preparing to redesign after years of sales. The new version of the ACM is still subject to a number of tests to determine the reliability of the product before it goes on the market. Because there are so many features, each test dataset can only detect several of them. Of course, the product must be available after all features have been tested. Since each test has time and material costs, they like to do the test as less as possible. Assume that running each test dataset costs the same, your job is finding the minimum number of test datasets that can cover the test of all features. For example, if there are 5 features that need to be tested, and there are 6 test datasets each can cover the features as follows:

- Test dataset a: 1
- Test dataset b: 2, 5
- Test dataset c: 2, 3, 4
- Test dataset d: 1, 3, 5
- Test dataset e: 1, 3, 4
- Test dataset f: 3, 5

Although $\{a, b, c\}$ may do the job, but $\{c, d\}$ will do the job better in the way of saving time and money.

Input Format

The first line of the input file contains one positive integer T representing the number of machines. For each machine, the first line consists of two integers n and m representing the features of machine that has to be tested and the number of test datasets. It follows by m lines, each line has a binary string of length n, showing whether the features can be detected by the test dataset or not (1 means yes, 0 means no).

Output Format

Output T lines. Each of them should be the minimum number of test dataset needed to test all features for that machine. If it is not possible to test all functions for the machine, output `-1`.

Technical Specification

- The number of machines $0 < T \leq 10$
- The number of functions to be tested $0 < n \leq 500$
- The number of test data $0 < m \leq 15$

Sample Input 1

```
5
3 3
100
011
111
5 6
10000
01001
01110
00111
10110
00101
6 7
000010
011000
100100
001000
000010
010000
110001
7 6
1001001
1001000
0001101
0010110
0110011
0100001
2 1
01
```

Sample Output 1

```
1
2
4
3
-1
```

Problem K
Length of Bundle Rope

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Due to the development of online shopping, the logistics industry which is highly connected with goods shipping has been so prosperous that the great amount of employees is needed. Therefore, Alex, a truck driver in this growing industry, was supposed to transport several parcels scattering in the warehouse to other cities in his daily routine.

According to the official safety requirements to the trucks running in the highway, Alex had to tie up all the packages tightly so that he could settle the goods safely on his truck. Alex knew that the length of the cords needed for bundling the packages on the truck was based on the size of the packages themselves. Also, n packages can be tied up well after $n-1$ bundles. Moreover, when bundling goods, Alex could only bundle two packages at one time to avoid scattering. Since the daily consumption of the cord was great and Alex was supposed to pay for it, he hopes to bundle all the goods with the shortest cord.

For example, there are 4 parcels in the size of 8, 5, 14, and 26 respectively. If Alex binds the first two together, the needed rope will be in the length of 13 ($8+5=13$) while the needed rope for the latter two packages will be 40 ($14+26=40$). If Alex keeps bundling these two items, the rope length he needs will be 53 ($13+40=53$). So the total length of the 4 packages will be 106 ($13+40+53=106$). If Alex tries another way by bundling the first two ($8+5=13$), adding up the third one ($13+14=27$), and then bundling the last item ($27+14=53$), he will only need the cord in the length of 93 ($13+27+53=93$). Now your mission is to help Alex finding the minimum length of the needed cord.

Input Format

The first line contains an integer T indicating the number of test cases. Each test case contains two lines. The first one contains a positive integer n indicating the number of packages. The second one contains n positive integers separated by a space to indicate the size of each parcel.

Output Format

For each test case, output the minimum length of the bundle rope required to tie all parcels together in one line.

Technical Specification

- $1 \leq T \leq 10$
- $1 \leq n \leq 1000$
- The size of each parcel is at most 1000.

Sample Input 1

```
4
6
2 3 4 4 5 7
5
5 15 40 30 10
10
3 1 5 4 8 2 6 1 1 2
9
3 2 1 6 5 2 6 4 3
```

Sample Output 1

```
63
205
100
98
```

Problem L
Largest Quadrilateral

Time limit: 6 seconds
Memory limit: 1024 megabytes

Problem Description

Given some points on a 2D Euclidean plane, please calculate the maximum area of quadrilaterals with vertices in given points. For example, points $A(0,0)$, $B(1,0)$, $C(3,1)$, $D(1,2)$, $E(0,1)$ are given. These points form 5 simple quadrilaterals $ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$ with areas 3, 2, 1.5, 3, 3 respectively and 10 complex quadrilaterals $ABDC$, $ABEC$, $ABED$, $ACED$, $BCED$, $ACBD$, $ACBE$, $ADBE$, $ADCE$, $BDCE$ with smaller areas. So the maximum area is 3.

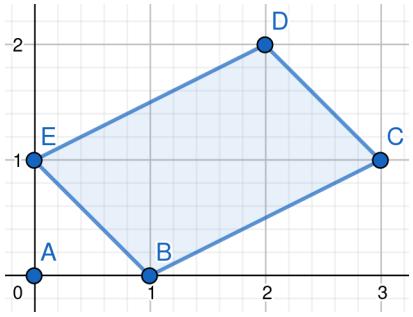

Figure 4: $BCDE$ has the maximum area 3 among all quadrilaterals.

Please notice that duplicated points may appear in the given points. All the degenerate cases are also taken as quadrilaterals such as quadrilateral $ABCD$ with $A(0,0)$, $B(0,0)$, $C(0,0)$, $D(0,0)$.

Input Format

The first line is an integer T indicating the number of test cases. The first line of each test case contains an integer N followed by N lines. Each line of the following N lines contains two integers X and Y representing a point (X, Y).

Output Format

For each test case, please output the maximum area among all quadrilaterals with vertices in given points.

Technical Specification

- $1 \leq T \leq 3$
- $4 \leq N \leq 4096$
- $0 \leq X \leq 10^9$
- $0 \leq Y \leq 10^9$
- You may not output numbers with scientific notaions. I.e., outputting `3E8` for `300000000` is not allowed.

- The area must be outputted without any redundant characters. I.e., outputting `3.0` for `3` is incorrect.

Sample Input 1

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

Sample Output 1

```
3
6
0
```

Sample Input 2

```
1
4
0 0
1 0
0 1
3 2
```

Sample Output 2

```
2.5
```

Problem M
DivModulo

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Modulo (mod) is a very common operator on integers. For two integers n and d with $d > 0$, $r \equiv (n \bmod d)$ is defined where $0 \leq r < d$ and there exists an integer q, such that $n = q \times d + r$. For example, $(200 \bmod 5) \equiv 0$ means that the remainder of 200 divided by 5 is 0. Here is another new operator called DivModulo (dmod) defined as follows. For two integers n and d with $d > 0$, $r \equiv (n \operatorname{dmod} d)$ is defined where there exists two integers m and h, such that $r \equiv (m \mod d)$, $n = m \times d^h$, and d is not a factor of m. For example, $(200 \operatorname{dmod} 5) \equiv 3$, since $(200 \operatorname{dmod} 5) \equiv (8 \times 5^2 \operatorname{dmod} 5) \equiv (8 \bmod 5) \equiv 3$.

Consider the factorials and the combination function. For an integer $M \geq 0$, the factorial $M!$ is defined as $M! = M \times (M-1) \times (M-2) \times \cdots \times 3 \times 2 \times 1$, and $0! = 1$ is defined. For integers M and N with $0 \leq N \leq M$, the combination function $C(M, N)$ is defined as $C(M, N) = \frac{M!}{N! \times (M-N)!}$. Now given three integers M, N, D with $D > 0$, please compute $C(M, N) \operatorname{dmod} D$. For example, $(C(9, 2) \operatorname{dmod} 3) \equiv (36 \operatorname{dmod} 3) \equiv (4 \times 3^2 \operatorname{dmod} 3) \equiv (4 \bmod 3) \equiv 1$.

Input Format

Three integers M, N and D are given in one line.

Output Format

Please output $C(M, N) \operatorname{dmod} D$ in one line.

Technical Specification

- $1 \leq M \leq 4 \times 10^{18}$
- $0 \leq N \leq M$
- $2 \leq D \leq 1.6 \times 10^7$

Sample Input 1

```
9 2 3
```

Sample Output 1

```
1
```

Sample Input 2

```
5 2 5
```

Sample Output 2

```
2
```

Sample Input 3

```
6 3 6
```

Sample Output 3

```
2
```

Sample Input 4

```
7654321 1234567 1050
```

Sample Output 4

```
210
```