Contents

Problem A. Nurse Schedule Listing

Input File Name:	`NrsSchedList.inp`
Output File Name:	`NrsSchedList.out`
Time Limit:	10s
Memory Limit:	128 MB

The director of a hospital want to schedule a working plan for a nurse in a given period of N consecutive days 1,..., N. Due to the policy of the hospital, each nurse cannot work all the days 1,..., N. Instead, there must be days off in which the nurse need to take a rest. A working plan is a sequence of disjoint working periods. A working period of a nurse is defined to be a sequence of consecutive days on which the nurse must work and the length of the working period is the number of consecutive days of that working period. The hospital imposes two constraints:

- Each nurse can take a rest only one day between two consecutive working periods. it means that if the nurse takes a rest today, then she has to work tomorrow (1)
- The length of each working period must be greater or equal to K_1 and less than or equal to K_2 (2)

A working plan is represented by a binary sequence (bit 1 means day on, and bit 0 means day off).

Given a positive integer K, the director of the hospital want to know the K^{th} working plan in the lexicographic order satisfying above constraints.

Input

The input consists of one line which contains 4 positive integers $N, K_1, K_2, K (N \leq 200, K_1 < K_2 \leq 70)$

Output

The output consists of one line containing the binary sequence describing the K^{th} working plan satisfying the above constraints. If there is not such K^{th} working plan, then output -1.

Example

NrsSchedList.inp	NrsSchedList.out
6 2 3	011011 110110 110111 111011

Explanation

There are 4 working plans described as follows

working plan 1	off	on	on	off	on	on
working plan 2	on	on	off	on	on	off
working plan 3	on	on	off	on	on	on
working plan 4	on	on	on	off	on	on

Problem B. Balanced Courses Assignment

Input File Name:	`BCA.inp`
Output File Name:	`BCA.out`
Time Limit:	60 s
Memory Limit:	128 MB

At the beginning of the semester, the head of a computer science department D have to assign courses to teachers in a balanced way. The department D has m teachers $T = \{1, 2, ..., m\}$ and n courses $C = \{1, 2, ..., n\}$. Each teacher $t \in T$ has a preference list which is a list of courses he/she can teach depending on his/her specialization. We known a list of pairs of conflicting two courses that cannot be assigned to the same teacher as these courses have been already scheduled in the same slot of the timetable. The load of a teacher is the number of courses assigned to her/him. How to assign n courses to m teacher such that each course assigned to a teacher is in his/her preference list, no two conflicting courses are assigned to the same teacher, and the maximal load is minimal.

Input

The input consists of following lines

- Line 1: contains two integer m and n ($1 \leq m \leq 10$, $1 \leq n \leq 30$)
- Line i+1: contains an positive integer k and k positive integers indicating the courses that teacher i can teach ($\forall i = 1, \ldots, m$)
- Line $m + 2$: contains an integer k
- Line $i + m + 2$: contains two integer i and j indicating two conflicting courses ($\forall i = 1, \ldots, k$)

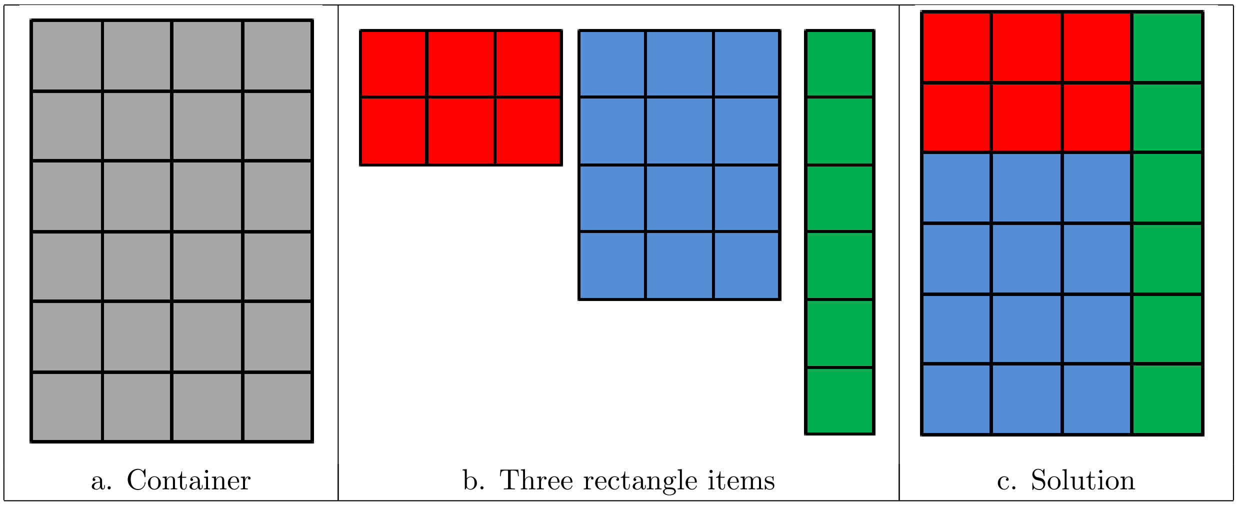
Output

The output contains a unique number which is the maximal load of the teachers in the solution found and the value -1 if not solution found.

Example

BCA.inp	BCA.out

4 12
5 1 3 5 10 12
5 9 3 4 8 12
6 1 2 3 4 9 7
7 1 2 3 5 6 10 11
25
1 2
1 3
1 5
2 4
2 5
2 6
3 5
3 7
3 10
4 6
4 9
5 6
5 7
5 8
6 8
6 9
7 8
7 10
7 11
8 9
8 11
8 12
9 12
10 11
11 12
	3


a. Container

b. Three rectangle items

c. Solution

Problem C. Container 2D

Input File Name:	STANDARD INPUT
Output File Name:	STANDARD OUTPUT
Time Limit:	60s
Memory Limit:	128 MB

There is a container having horizontal size W and vertical size H. There are N rectangle items 1, 2, ..., N in which item i has horizontal size w_i and vertical size h_i. Find the way to place these N items into the container such that

- The sides of items are packed in parallel with the sides of the container
- The items cannot rotated
- No two items overlap.

Input

The input consists of following lines

- Line 1: contains two integer H and W ($1 \leq H, W \leq 30$)
- Line 2: contains N ($1 \leq N \leq 12$)
- Line $i + 2$ ($\forall i = 1, \ldots, N$): contains two integers h_i and w_i

Output

The output contains a unique number 0 (if we cannot place items) or 1 (if we can place items)

Example

STANDARD INPUT	STANDARD OUTPUT

6 4
3
2 3
6 1
4 3
	1

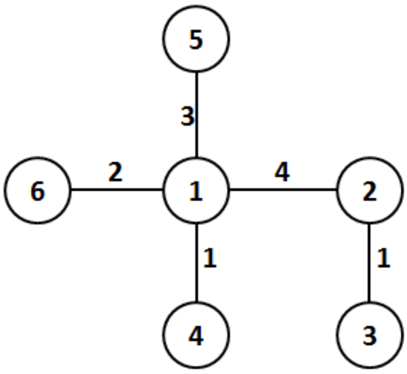


Figure 1: Communication networks G

Problem D. Networks

Input File Name:	`STANDARD INPUT`
Output File Name:	`STANDARD OUTPUT`
Time Limit:	1 s
Memory Limit:	128 MB

The network administrator of a company have to analyze the current state of their communication network all over the world. The communication network consists of servers and cable links between these servers, each link has a cost. A k-route is a sequence of $k+1$ different servers in which two consecutive servers are connected by a cable link. A cycle is a k-route (for any $k > 1$) such that the beginning and the terminating servers are connected by a cable link. The communication network contains no cycle. The cost of a k-route is the sum of cost of links between two consecutive servers of the k-route. One of the indicators of the analysis is the k-route having minimal cost of the network for a given value of k.

The 2-route having minimal cost of the communication network in Figure 1 is 6-1-4 with the cost 3.

Given the communication network G and an integral value k, help the network administrator to find the k-route having minimal cost of G.

Input

The input consists of following lines

- Line 1: contains two integer n and k ($1 \leq n \leq 10000$, $1 \leq k \leq 2000$) in which n is the number of servers of the communication network G (servers are numbered 1,2,...,n)
- Line 2: contains an integer m ($1 \leq m \leq 10000$) which is the number of cable links between servers of G
- Line $i+2$: contains three integers u, v, and w: u and v are two end points of the i^{th} link of G ($\forall i = 1, \ldots, m$), w is the cost of this link.

Output

The output contains the cost of the k-route found.

Example

STANDARD INPUT	STANDARD OUTPUT

6 2
5
1 2 4
1 4 1
1 5 3
1 6 2
2 3 1
	3

Problem E. Money Changing

Input File Name:	`change.inp`
Output File Name:	`change.out`
Time Limit:	1 s
Memory Limit:	128 MB

Minh go shopping at the SS shop. The shop has currency denominations: 1\$, 5\$, 10\$, 50\$, 100\$, 500\$. Minh takes some items at the shop and pay an amount of 1000\$. Your task to devise a method to pay back amount to customer using fewest number of money notes.

Input

The input consists of only one single integer N $(1 \leq N \leq 999)$ denoting the total value of the taken items.

Output

The output consists of only one single integer denoting the number of money notes.

Example

change.inp	change.out
380	4

Explanation

The shop has to pay back 620\$ by giving 1 paper of 500\$, 1 paper of 100\$ and 2 papers of 10\$.

Problem F. ATM withdrawal

Input File Name:	`atm.inp`
Output File Name:	`atm.out`
Time Limit:	1 s
Memory Limit:	256 MB

Vinh works for an ATM machine manufacturing company. The basic functionality of an ATM machine is cash withdrawal. When a user requests a cash withdrawal of W VND (Vietnamese Dong), the ATM has to dispense N money notes such that they sum up to W. For the next generation of ATM machine, Vinh is working on an algorithm to minimize the number N of money notes for each cash withdrawal transaction.

Your task is to help Vinh to do his job given that the money notes come in the values of $1000, 2000, 3000, 5000, 1000 \cdot 10^1, 2000 \cdot 10^1, 3000 \cdot 10^1, 5000 \cdot 10^1, \ldots, 1000 \cdot 10^c, 2000 \cdot 10^c, 3000 \cdot 10^c, 5000 \cdot 10^c$ where c is a positive integer and Vinh has unlimited supply of money notes for each value.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 1000. The following lines describe the datasets.

- The first line consists of one positive integer W ($W \leq 10^{18}$);
- The second line consists of one positive integer c ($c \leq 15$).

Output

For each dataset, write in one line two space-separated integers N and S where S is the number of ways to dispense the fewest number N of money notes. In case there is no way to serve the cash withdrawal request, write out 0 in one line instead.

Examples

atm.inp	atm.out
2 1000 1 7000 1	1 1 2 1

Problem G. Planting Trees

Input File Name:	`ptrees.inp`
Output File Name:	`ptrees.out`
Time Limit:	1 s
Memory Limit:	256 MB

Farmer Jon has recently bought n tree seedlings that he wants to plant in his yard. It takes 1 day for Jon to plant a seedling[1], and for each tree Jon knows exactly in how many days after planting it grows to full maturity. Jon would also like to throw a party for his farmer friends, but in order to impress them he would like to organize the party only after all the trees have grown. More precisely, the party can be organized at earliest on the next day after the last tree has grown up.

Help Jon to find out when is the earliest day when the party can take place.
Jon can choose the order of planting the trees as he likes, so he wants to plant the trees in such a way that the party will be as soon as possible.

Input

The input consists of two lines. The first line contains a single integer N ($1 \leq N \leq 100\,000$) denoting the number of seedlings. Then a line with N integers t_i follows ($1 \leq t_i \leq 1\,000\,000$), where t_i denotes the number of days it takes for the ith tree to grow.

Output

You program should output exactly one line containing one integer, denoting the earliest day when the party can be organized. The days are numbered $1, 2, 3, \ldots$ beginning from the current moment.

Examples

ptrees.inp	ptrees.out
4 2 3 4 3	7
6 39 38 9 35 39 20	42

[1] Jon isn't particularly hardworking.

Problem H. Pie

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

My birthday is coming up and traditionally Im serving pie. Not just one pie, no, I have a number N of them, of various tastes and of various sizes. F of my friends are coming to my party and each of them gets a piece of pie. This should be one piece of one pie, not several small pieces since that looks messy. This piece can be one whole pie though. My friends are very annoying and if one of them gets a bigger piece than the others, they start complaining. Therefore all of them should get equally sized (but not necessarily equally shaped) pieces, even if this leads to some pie getting spoiled (which is better than spoiling the party). Of course, I want a piece of pie for myself too, and that piece should also be of the same size. What is the largest possible piece size all of us can get? All the pies are cylindrical in shape and they all have the same height 1, but the radii of the pies can be different.

Input

One line with a positive integer: the number of test cases. Then for each test case:

- One line with two integers N and F with $1 \leq N, F \leq 10000$: the number of pies and the number of friends.
- One line with N integers r_i with $1 \leq r_i \leq 10000$: the radii of the pies.

Output

For each test case, output one line with the largest possible volume V such that me and my friends can all get a pie piece of size V. The answer should be given as a floating point number rounding to 6 digits after the floating point.

Examples

stdin	stdout

3
3 3
4 3 3
1 24
5
10 5
1 4 2 3 4 5 6 5 4 2
	25.132741
	3.141593
	50.265482

Problem I. Fibonacci Words

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

The Fibonacci word sequence of bit strings is defined as:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

Here denotes concatenation of strings. The first few elements are:

n	F(n)
0	0
1	1
2	10
3	101
4	10110
5	10110101
6	1011010110110
7	101101011011010110101
8	1011010110110101101011011010110110
9	1011010110110101101011011010110110101101011011010110101

Given a bit pattern p and a number n, how often does p occur in $F(n)$?

Input

The first line of each test case contains the integer n ($0 \leq n \leq 100$). The second line contains the bit pattern p. The pattern p is nonempty and has a length of at most 100 000 characters.

Output

For each test case, display its case number followed by the number of occurrences of the bit pattern p in $F(n)$. Occurrences may overlap. The number of occurrences will be less than 2^{63}.

Examples

stdin	stdout
6 10	Case 1: 5

Problem J. The Hamming Distance

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1s
Memory Limit:	256 MB

The Hamming distance between two strings of bits (binary integers) is the number of corresponding bit positions that differ. This can be found by using XOR on corresponding bits or equivalently, by adding corresponding bits (base 2) without a carry. For example, in the two bit strings that follow:

A	0 1 0 0 1 0 1 0 0 0
B	1 1 0 1 0 1 0 1 0 0
A XOR B =	1 0 0 1 1 1 1 1 0 0

The Hamming distance (H) between these 10-bit strings is 6, the number of 1s in the XOR string.

Input

Input consists of several datasets. The first line of the input contains the number of datasets, and its followed by a blank line. Each dataset contains N, the length of the bit strings and H, the Hamming distance, on the same line. There is a blank line between test cases.

Output

For each dataset print a list of all possible bit strings of length N that are Hamming distance H from the bit string containing all 0s (origin). That is, all bit strings of length N with exactly H 1s printed in ascending lexicographical order.

The number of such bit strings is equal to the combinatorial symbol $C(N, H)$. This is the number of possible combinations of NH zeros and H ones. It is equal to

$$\frac{N!}{(NH)!H!}$$

This number can be very large. The program should work for $1 \leq H \leq N \leq 16$.

Print a blank line between datasets.

Examples

stdin	stdout
2	0011
4 2	0101
	0110
1 0	1001
	1010
	1100
	0

Problem K. Route Planning

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1s
Memory Limit:	256 MB

Superior Island is a very picturesque island and only bicycles are allowed on the island. Therefore, there are many one-way bicycle roads connecting the different best photo-shooting spots on the island. To help the visitors plan their trip to the island, the tourism commission wants to designate r different bicycle routes that go through some of the best photo-shooting spots on the island. Given a map of all the bicycle roads on the island and a list of the best photo-shooting spots to be included on each of the three planned routes (non-listed spots must not be included in the route), please write a program to plan each of the r routes so that the distance on each route is minimal. Note that each best photo-shooting spot may only appear at most once on the route.

Input

There are two parts to the input. The first part of input gives the information of the bicycle roads on the island. The first line contains two integer n and r , $n \leq 100$ and $r \leq 10$, indicating that there are n best photo-shooting spots on the island and there are r routes to be planned. The next n lines (line 2 through line $n + 1$) contains $n \times n$ integers (n lines with n integers on each line), where the j -th integer on line i denotes the distance from best photo-shooting spot $i - 1$ to best photo-shooting spot j ; the distances are all between 0 and 10, where 0 indicates that there is no one-way road going from best photo-shooting spot $i - 1$ to spot j.

The second part of input has r lines, denoting the r sightseeing routes to be planed. Each line lists the best photo-shooting stops to be included in that route. The integers on each line denote the recommended photo-shooting stops on that particular sightseeing route. The first integer on the line is the starting point of the route and the last integer is the last stop on the route. However, the stops in between can be visited in any order.

Output

Output r integers on r lines (one integer per line) indicating the distance of each of the r planned routes. If a route is not possible, output ‘0’.

Examples

stdin	stdout

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

Problem L. The Tower of Babylon

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 giy
Memory Limit:	256 MB

Perhaps you have heard of the legend of the Tower of Babylon. Nowadays many details of this tale have been forgotten. So now, in line with the educational nature of this contest, we will tell you the whole story:

The babylonians had n types of blocks, and an unlimited supply of blocks of each type. Each type-i block was a rectangular solid with linear dimensions (x_i, y_i, z_i) . A block could be reoriented so that any two of its three dimensions determined the dimensions of the base and the other dimension was the height. They wanted to construct the tallest tower possible by stacking blocks. The problem was that, in building a tower, one block could only be placed on top of another block as long as the two base dimensions of the upper block were both strictly smaller than the corresponding base dimensions of the lower block. This meant, for example, that blocks oriented to have equal-sized bases couldn't be stacked.

Your job is to write a program that determines the height of the tallest tower the babylonians can build with a given set of blocks.

Input

The input file will contain one or more test cases. The first line of each test case contains an integer n, representing the number of different blocks in the following data set. The maximum value for n is 30. Each of the next n lines contains three integers representing the values x_i, y_i and z_i.

Input is terminated by a value of zero (0) for n.

Output

For each test case, print one line containing the case number (they are numbered sequentially starting from 1) and the height of the tallest possible tower in the format "Case *case*: maximum height = height"

Examples

stdin	stdout
`1`	`Case 1: maximum height = 40`
`10 20 30`	`Case 2: maximum height = 21`
`2`	`Case 3: maximum height = 28`
`6 8 10`	`Case 4: maximum height = 342`
`5 5 5`	
`7`	
`1 1 1`	
`2 2 2`	
`3 3 3`	
`4 4 4`	
`5 5 5`	
`6 6 6`	
`7 7 7`	
`5`	
`31 41 59`	
`26 53 58`	
`97 93 23`	
`84 62 64`	
`33 83 27`	
`0`	

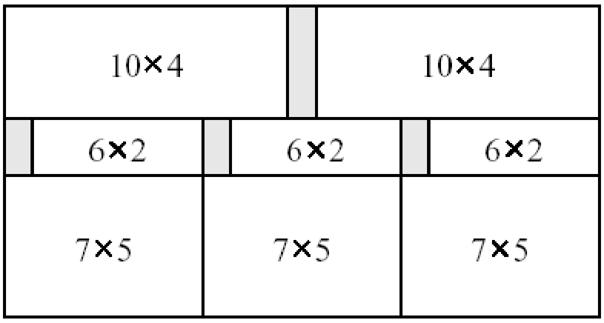
Problem M. Marble Cut

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1s
Memory Limit:	256 MB

Famous sculptor Phong is making preparations to build a marvelous monument. For this purpose he needs rectangular marble plates of sizes $W_1 \times H_1, W_2 \times H_2, \dots, W_N \times H_N$.

Recently, Phong has received a large rectangular marble slab. He wants to cut the slab to obtain plates of the desired sizes. Any piece of marble (the slab or the plates cut from it) can be cut either horizontally or vertically into two rectangular plates with integral widths and heights, cutting completely through that piece. This is the only way to cut pieces and pieces cannot be joined together. Since the marble has a pattern on it, the plates cannot be rotated: if Phong cuts a plate of size $A \times B$ then it cannot be used as a plate of size $B \times A$ unless $A = B$. He can make zero or more plates of each desired size. A marble plate is wasted if it is not of any of the desired sizes after all cuts are completed. Phong wonders how to cut the initial slab so that as little of it as possible will be wasted.

As an example, assume that in the figure below the width of the original slab is 21 and the height of the original slab is 11, and the desired plate sizes are $10 \times 4, 6 \times 2, 7 \times 5$, and 15×10. The minimum possible area wasted is 10, and the figure shows one sequence of cuts with total waste area of size 10.



Your task is to write a program that, given the size of the original slab and the desired plate sizes, calculates the minimum total area of the original slab that must be wasted.

Input

- The first line of input contains two integers: first W, the width of the original slab, and then H, the height of the original slab;
- The second line contains one integer N: the number of desired plate sizes. The following N lines contain the desired plate sizes. Each of these lines contains two integers: first the width W_i and then the height H_i of that desired plate size ($1 \leq i \leq N$).

Output

For each dataset, write in one line a single integer: the minimum total area of the original slab that must be wasted.

Examples

stdin	stdout

1
21 11
4
10 4
6 2
7 5
15 10
	10

Scoring

In all datasets, $1 \leq W \leq 600, 1 \leq H \leq 600, 0 < N \leq 200, 1 \leq W_i \leq W$, and $1 \leq H_i \leq H$. Additionally, in 50% of the inputs, $W \leq 20, H \leq 20$ and $N \leq 5$.

Problem N. Gold

Input File Name:	`Gold.inp`
Output File Name:	`Gold.out`
Time Limit:	1 s
Memory Limit:	128 MB

The Kingdom ALPHA has n warehouses of golds located on a straight line and are numbered 1, 2, ..., n. The warehouse i has amount of a_i (a_i is non-negative integer) and is located at coordinate i ($\forall i = 1, \ldots, n$). The King of ALPHA opens a competition for hunters who are responsible to find a subset of gold warehouses having largest total amount of golds with respect to the condition that the distance between two selected warehouses must be greater than or equal to L_1 and less than or equal to L_2.

Input

The input consists of following lines:

- Line 1 contains n, L_1, and L_2 ($1 \leq n \leq 100000, 1 \leq L_1 \leq L_2 \leq n$)
- Line 2 contains n integers $a_1, a_2, \ldots, a_n$

Output

The output consists of T lines, line t contains only one single integer denoting the total amount of golds of selected warehouses of the test t ($t = 1, 2, \ldots, T$).

Example

`Gold.inp`	`Gold.out`

2
6 2 3
3 5 9 6 7 4
10 2 8
1 1 1 2 0 0 0 2 2 1
	19
	6

Explanation

‘

Problem O. Nurse

Input File Name:	`Nurse.inp`
Output File Name:	`Nurse.out`
Time Limit:	1 s
Memory Limit:	128 MB

The director of a hospital want to schedule a working plan for a nurse in a given period of N consecutive days 1,..., N. Due to the policy of the hospital, each nurse cannot work all the days 1,..., N. Instead, there must be days off in which the nurse need to take a rest. A working plan is a sequence of disjoint working periods. A working period of a nurse is defined to be a sequence of consecutive days on which the nurse must work and the length of the working period is the number of consecutive days of that working period. The hospital imposes two constraints:

- Each nurse can take a rest only one day between two consecutive working periods. it means that if the nurse takes a rest today, then she has to work tomorrow (1)
- The length of each working period must be greater or equal to K_1 and less than or equal to K_2 (2)

The director of the hospital want to know how many possible working plans satisfying above constraint?

Input

The input consists of one line which contains 3 positive integers $N, K_1, K_2 (N \leq 1000, K_1 < K_2 \leq 400)$

Output

The output consists of only one single integer M modulo $10^9 + 7$ where M is the total working plans satisfying the above constraints.

Example

Nurse.inp	Nurse.out
6 2 3	4

Explanation

There are 4 working plans described as follows

working plan 1	on	on	off	on	on	on
working plan 2	on	on	off	on	on	off
working plan 3	off	on	on	off	on	on
working plan 4	on	on	on	off	on	on

Problem P. Machine

Input File Name:	`STANDARD INPUT`
Output File Name:	`STANDARD OUTPUT`
Time Limit:	1 s
Memory Limit:	128 MB

An engineer needs to schedule a machine to run on some given periods $1, \ldots, n$ to produce a chemical product $\mathcal{C}$. Each period i is represented by a starting time point s_i and terminating time point t_i ($s_i < t_i$). Due to a technical constraint, the machine must run on exactly two periods that are not overlap (two periods i and j are not overlap if $t_i < s_j$ or $t_j < s_i$). If the machine is runned on the period i, then the amount of $\mathcal{C}$ it will produce is equal to the duration of the period i (which is equal to $t_i - s_i$). Help the engineer to select two not-overlap periods to run the machine such that the amount of $\mathcal{C}$ produced is maximal.

Input

The input consists the following lines:

- Line 1: contains the positive integer n ($2 \leq n \leq 10^6$)
- Line $i + 1$: contains two positive integer s_i and t_i ($1 \leq s_i < t_i \leq 10^6$)

Output

The output consists of only one single integer which is the amount of product $\mathcal{C}$ the machine will produce in the two selected periods. In case there is no solution (there does not exist two periods that are not overlap), the output contains the value -1.

Example

STANDARD INPUT	STANDARD OUTPUT

5
8 12
6 11
3 9
2 5
1 4
	8

Explanation

The machine will be runned on two periods [2, 5] and [6, 11] and produce 8 unit of product $\mathcal{C}$.

Problem Q. Phone List

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

Given a list of phone numbers, determine if it is consistent in the sense that no number is the prefix of another. Lets say the phone catalogue listed these numbers:

- Emergency 911
- Alice 97 625 999
- Bob 91 12 54 26

In this case, its not possible to call Bob, because the central would direct your call to the emergency line as soon as you had dialled the first three digits of Bobs phone number. So this list would not be consistent.

Input

The first line of input gives a single integer, $1 \leq t \leq 40$, the number of test cases. Each test case starts with n, the number of phone numbers, on a separate line, $1 \leq n \leq 10000$. Then follows n lines with one unique phone number on each line. A phone number is a sequence of at most ten digits.

Output

For each test case, output ‘YES if the list is consistent, or ‘NO otherwise.

Examples

stdin	stdout
2	NO
3	YES
911	
97625999	
91125426	
5	
113	
12340	
123440	
12345	
98346	

Problem R. DNA Repetitions

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

The Institute of Bioinformatics and Medicine (IBM) of your country has been studying the DNA sequences of several organisms, including the human one. Before analyzing the DNA of an organism, the investigators must extract the DNA from the cells of the organism and decode it with a process called "sequencing". A technique used to decode a DNA sequence is the "shotgun sequencing". This technique is a method applied to decode long DNA strands by cutting randomly many copies of the same strand to generate smaller fragments, which are sequenced reading the DNA bases (A, C, G and T) with a special machine, and re-assembled together using a special algorithm to build the entire sequence. Normally, a DNA strand has many segments that repeat two or more times over the sequence (these segments are called "repetitions"). The repetitions are not completely identified by the shotgun method because the re-assembling process is not able to differentiate two identical fragments that are substrings of two distinct repetitions. The scientists of the institute decoded successfully the DNA sequences of numerous bacterias from the same family, with other method of sequencing (much more expensive than the shotgun process) that avoids the problem of repetitions. The biologists wonder if it was a waste of money the application of the other method because they believe there is not any large repeated fragment in the DNA of the bacterias of the family studied. The biologists contacted you to write a program that, given a DNA strand, finds the largest substring that is repeated two or more times in the sequence.

Input

The first line of the input contains an integer T specifying the number of test cases ($1 \leq T \leq 100$). Each test case consists of a single line of text that represents a DNA sequence S of length n ($1 \leq n \leq 1000$). You can suppose that each sequence S only contains the letters 'A', 'C', 'G' and 'T'.

Output

For each sequence in the input, print a single line specifying the largest substring of S that appears two or more times repeated in S, followed by a space, and the number of ocurrences of the substring in S. If there are two or more substrings of maximal length that are repeated, you must choose the least according to the lexicographic order. If there is no repetition in S, print 'No repetitions found!'.

Examples

stdin	stdout
`6`	`A 3`
`GATTACA`	`GAGAG 2`
`GAGAGAG`	`GATTACA 2`
`GATTACAGATTACA`	`No repetitions found!`
`TGAC`	`T 2`
`TGTAC`	`A 2`
`TTGGAACC`	

Problem S. DNA Sequences

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

A DNA molecule consists of two strands that wrap around each other to resemble a twisted ladder whose sides, made of sugar and phosphate molecules, are connected by rungs of nitrogen-containing chemicalscalledbases. Each strand is a linear arrangement of repeating similar units called nucleotides, which are each composed of one sugar, one phosphate, and a nitrogenous base. Four different bases are present in DNA: adenine (A), thymine (T), cytosine (C), and guanine (G). The particular order of the bases arranged along the sugar-phosphate backbone is called the DNA sequence; the sequence specifies the exact genetic instructions required to create a particular organism with its own unique traits.

Geneticists often compare DNA strands and are interested in finding the longest common base sequence in the two strands. Note that these strands can be represented as strings consisting of the letters a, t, c and g. So, the longest common sequence in the two strands atgc and tga is tg. It is entirely possible that two different common sequences exist that are the same length and are the longest possible common sequences. For example in the strands atgc and gctg, the longest common sequences are gc and tg.

Your task is to write a program that accepts as input two strings representing DNA strands, and prints as output the longest common sequence(s) in lexicographical order.

Input

The input file contains several test cases with a blank line between two consecutive. The strings are at most 300 characters-long.

Output

Output For each test case, print all the longest common sequences, one per line, in lexicographical order. If there isn't any common sequence between the two strings, just print: 'No common sequence.' Print a blank line between the output of consecutive datasets.

Examples

stdin	stdout

`atgc`
`tga`
`atgc`
`gctg`
	`tg`
	`gc`
	`tg`

Problem T. InterCity Bus

Input File Name:	`stdinp`
Output File Name:	`stdout`
Time Limit:	1s
Memory Limit:	256 MB

Country SS consists of N towns indexed from 1 to N, and each town i has its own inter city bus (IC Bus for short) system i. There is K roads between towns, each road connects two different towns. The bus can move freely in both directions on the road.

Quang is living in the town 1 in the country, and decided to go to the grandmother's house in the town N by some inter city buses. There are some special rules in this country:

- If the passenger want to use the IC Bus of the town i, he has to only ride at the town i.
- The bus fares of the IC Bus system i is C_i regardless of the distance that the passenger used.
- The IC Bus system i allows to pass maximum D_i towns per trip. If the trip has to pass more than D_i towns, the passenger has to change to another IC Bus system.
- The passenger will not be able ride to or down from the bus at a middle point different than the town.

Your task is to to find the minimum value of the sum of the fare needed for Quang to reach the town N from the town 1.

Input

The input consists of $1 + N + K$ lines.

The first line contains two positive integers N and K ($2 \leq N \leq 5000; N - 1 \leq K \leq 10000$).

i-th line in the N following lines contains 2 positive integers C_i and D_i ($1 \leq C_i \leq 10000; 1 \leq D_i \leq N$) which are the taxi fare and the maximum number of passing towns of the IC Bus system i.

Each line in the K following lines contains two positive integers i and j ($1 \leq i < j \leq N$) which means these two towns has a direct road connecting them.

Output

You should output on a single line an unique integer that is the minimum value of the sum of the fare necessary for Quang to go to the town N from the town 1.

Examples

stdinp	stdout

6 6
400 2
200 1
500 3
900 1
400 4
200 5
1 2
1 5
2 3
2 4
3 6
4 6
	800

Explanation

Quang uses the IC Bus of the town 1 and then of the town 5.

Problem U. Edges Adding

Input File Name:	`stdinp`
Output File Name:	`stdout`
Time Limit:	1 giy
Memory Limit:	256 MB

An is obseving an undirected graph. He wonders if there exits in this graph 2 vertices that connecting these 2 vertices will create exactly one more simple cycle. Recall that a simple cycle may be defined either as a sequence of vertices with no repetitions of vertices and edges allowed, other than the repetition of the starting and ending vertex, with each two consecutive vertices in the sequence adjacent to each other in the graph.

Your task is to help An count the number of pairs of vertices satisfying the conditions above.

Input

The first line consist of two positive integers n, m ($n, m \leq 10^5$) which are the number of vertices and the number of edges of the given graph.

Each line in the m following lines contains two positive integers u, v ($u, v \leq n$) which are two vertices connected by an edge.

Output

You should output on a single line an unique integer that is the number of pairs you found.

Examples

stdinp	stdout

5 4
1 2
2 3
3 4
4 5
	6

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

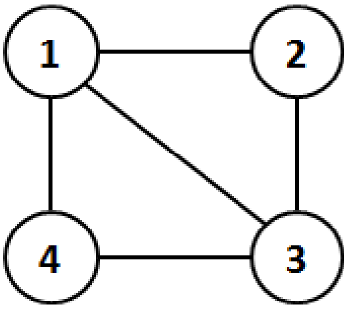


Figure 2: Undirected graph G

Problem V. KPath

Input File Name:	`STANDARD INPUT`
Output File Name:	`STANDARD OUTPUT`
Time Limit:	1s
Memory Limit:	128 MB

A k-path on a given undirected graph is a path having exactly k edges and containing no repeated nodes. Given an undirected graph G and an integral value k, count how many k-paths on G.

There six 3-paths on the graph in Figure 2 which are: 1-2-3-4, 2-3-4-1, 3-4-1-2, 4-1-2-3, 2-1-3-4, 4-1-3-2

Input

The input consists of following lines

- Line 1: contains two integer n and k ($1 \leq n \leq 30$, $1 \leq k \leq 10$) in which n is the number of nodes of the graph G (nodes are numbered 1,2,...,n)
- Line 2: contains an integer m ($1 \leq m \leq 60$) which is the number of edges of G
- Line $i+2$: contains two integers u and v representing two end points of the i^{th} edge of G ($\forall i = 1, \ldots, m$)

Output

The output contains the number of k-paths of G

Example

STANDARD INPUT	STANDARD OUTPUT

4 3
5
1 2
1 3
1 4
2 3
3 4
	6

Problem W. Longest Common Substring of n Strings

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 s
Memory Limit:	256 MB

A string P is call substring of the string S if P is a sequence that appears in the same order and necessarily contiguous in the string S.

Given n string sequences $S_1, S_2, \ldots, S_n$, your task is to write a program to find the length of longest substring present in all of them.

Input

The input file contains n lines, the line i contains the string S_i with uniquely upper-case letters. The total length of all n strings is not exceeded to 10^5.

Output

Print uniquely one integer which is the length of the longest common substring.

Examples

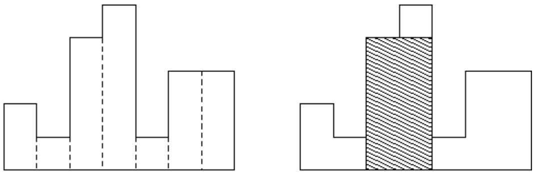
stdin	stdout
ABCXYZ XYZABC XYABCZ	3

Problem X. Lu Ban

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 giy
Memory Limit:	256 MB

Lu Ban was an ancient Chinese carpenter, engineer, and inventor. Once, the King invite Lu Ban to create a rectangular chess board from a precious marble.

The face of the marble can be described as a polygon joined by continuous small rectangular marbles. Two continuous small marbles have a common border. They can have different heights but all have the same width which is equal to one unit. In the following figure, the marble includes the rectangles with heights in a row from left to right 2, 1, 4, 5, 1, 3, 3 and the width is all 1.



Your task is to help Lu Ban the maximum area rectangle inside the marble. In the figure above, the answer is the cross-hatching rectangle.

Input

The input file contains one or more test cases. Each test case describes one polygon in one line. The line starts with a positive integer n ($n \leq 10^6$) which is the number of small rectangles joining to be the polygon. The next n integers in the line $l_1, l_2, \ldots, l_n$ where $0 \leq l_i \leq 10^8$ describes the heights of the small rectangles from left to right respectively. The file ends with the single 0.

Output

Each line contains the answer of the corresponding test case which is the area of your found rectangle.

Examples

stdin	stdout

`7 2 1 4 5 1 3 3`
`4 1000 1000 1000 1000`
`0`
	`8`
	`4000`

Problem Y. The Maximum Subsequence with Bounded Length

Input File Name:	`stdin`
Output File Name:	`stdout`
Time Limit:	1 giy
Memory Limit:	256 MB

Given an array of integers $A = a_1, a_2, \ldots, a_n$, a subsequence of A is a sequence of continuous elements in A, that means a sequence of form $a_i, a_{i+1}, \ldots, a_j$ $(1 \leq i \leq j \leq n)$. The length of the subsequence is the number of its elements. The weight of the subsequence is the sum of all elements. A subsequence has the bounded length if its length is greater or equal to L_1 and smaller or equal to L_2.

Your task is to find the maximum weight subsequence of A with length bounded by L_1 and L_2.

Input

The fist line contains 4 positive integers n, L_1, L_2 $(n \leq 10^6, L_1 \leq L_2 \leq n)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$.

Output

Print uniquely one integer which is the found weight.

Examples

stdin	stdout
6 3 4 3 5 -9 6 7 -4	9

The Maximum Subsequence with Bounded Length is 5, -9, 6, 7 having the weight 9.