Problem A. Ambitious journey

Source file name:	`ambitious.c, ambitious.cpp, ambitious.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

John the explorer is known to travel a lot, on each of his travels he plans always to collect the most souvenirs he can. John is not that greedy, He doesn't want to keep all the souvenirs for himself, his family is huge and He always wants to have enough souvenirs so He can distribute them into his family. On his last travels, he found that the souvenir stores are always ditributed into a square grid of size N, each coordinate of the grid has a store where he can buy up to $S_{i,j}$ souvenirs.

As John is very considered with his family, He always makes the duration of his travels the less time possible so he can spend more time with his family than traveling. To achieve this, He always lands on the coordinate $(1,1)$ and moves up to the coordinate (N,N), assuming John is in the coordinate (i,j), the next coordinate he will go is either $(i+1,j)$ or $(i,j+1)$, also, each time John arrives to a coordinate (including his landing in $(1,1)$) he will buy all the souvenir available in that store.

John has the maps of the next places He will be traveling. Help John writing a program to calculate for each map the maximum ammount of souvenirs he can buy.

Input

The input consist of several test cases. Each test case begins with a line with a single integer N, followed by N lines with N integer numbers each, where the $i-th$ line and $j-th$ column of the input is the value $S_{i,j}$). The end of the test cases is given by a line where N = 0 , this last line should not be processed as a test case.

- $1 \leq N \leq 1000$

Output

For each test case print in one line the maximum ammount of souvenirs John can get from his travel.

Example

Input	Output

`3`
`1 2 3`
`1 2 3`
`1 2 3`
`4`
`10 28 12 3`
`8 25 11 13`
`15 21 32 10`
`10 9 8 7`
`0`
	`12`
	`133`

Explication

For the first test case John can follow the path $(1,1),(1,2),(1,3),(2,3),(3,3)$ to sum up to 12 souvenirs, there is no path where He can get more souvenirs.

Problem B. Building lost

Source file name:	`building.c, building.cpp, building.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

On the Amazing City of Mexico (ACM) each street has up to N buildings, each building of the street has a number between 1 and N. If you were walking over the street, you will see that the numeration of the buildings is ordered, this is, the first building has the number 1, the second building has the number 2, and so on.

During the last ACM Programming Contest, John was instructed to give a tour to the foreign contestants who are visiting the town, in this travel he found that all the streets have a missing building, this is, there is a number X between 1 and N that no building in the street has that number, this as you can see, means the street has only $N - 1$ buildings not N.

John is preparing a letter to the government where he will ask they to fix this problem, if there are $N - 1$ buildings in the street they should numerate the buildings appropriately, He have colected the numbering from T streets. Since some streets have a large amount of buildings, it is difficult for him to find the missing building in all of them, that's why he is requesting your help to write a program that finds the missing building for each street.

Input

The first line of input is a single number T, followed by the description of the T streets. Each street description starts with a line with a single number N followed by a line with $N - 1$ numbers showing the numbers the buildings have.

- $1 \leq T \leq 100$
- $1 \leq N \leq 100,000$

Output

For each test case print in one line the number X missing in that street.

Example

Input	Output

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

Problem C. Counting trapezoids

Source file name:	`counting.c, counting.cpp, counting.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

In mathematics there are sets of interesting numbers, some of them have a geometric representation, some examples are the square numbers and the triangular numbers. Square numbers are those that if you had N units you can arrange them in such a way that you can create a square with that units. Triangular numbers are those where the N units can be arranged in such a way that a triangle is created from L consecutive numbers starting from 1.

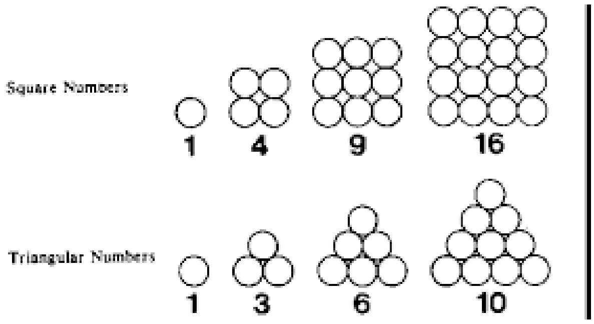

Figure 1: Some square and triangular numbers

There is another interesting set, we call it the trapezoid numbers, a trapezoid number N, is a number where the units can be arranged in a trapezoid figure from a number of 2 or more consecutive positive numbers, Triangular numbers are also trapezoid numbers that starts counting from 1. An example of trapezoid number is 5 which can be represented as a trapezoid with two numbers {2,3}.

Your task is given a number N , determine how many distinct trapezoids can be drawn using N units?

Input

The input consists of several test cases. Each test case consists of a single line containing a single number N. The end of the test cases is given by the end of file (EOF).

- $1 \leq N \leq 10^9$

Output

For each test case print in one line the number of different ways N can be represented as a trapezoid.

Example

Input	Output
1 3 9	0 1 2

Explication

There are 3 test cases in the file.
For the first test case the output is 0, there is no way to represent 1 as a trapezoid.
For the second test case the output is 1, the only way to represent 3 as a trapezoid is {1,2}
For the third test case the output is 2, there are two ways to represent 9 as a trapezoid :{2,3,4}, {4,5}.

Problem D. Dynamic Writing

Source file name:	`dynamic.c, dynamic.cpp, dynamic.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

Only one month to end school and professor teached a new way of writing... It's called dynamic writing, why dynamic? I don't know, but it basically consists on writing words separated by spaces (yes, pretty much as every one writes). Dynamic writing has some rules:

- A letter written with the same length of words separated by the same spaces is considered the same, i.e the letter "There_is_a_way" and "Never_is_a_one" (underscore represents a whitespace) are the same, since the lengths of each word separated by a space is the same.
- The length of a letter is the sum of the length of the words, in the previous example the length is 11 (5+2+1+3).
- There can be words with length 0, this is represented by consecutive whitespaces: "ABC__ABC" has 3 words, two with length 3 ("ABC") and 1 with length 0.
- Leading or trailing whitespaces also separate words "_ABC_" has 3 words, one at the beginning with length 0, then "ABC" with length 3, finally one at the end with length 0.

.

Since you like programming questions, you decided to give you a programming challenge with this new "dynamic writing" thing. Given two values N and K, how many different letters can be written that has length N and exactly K whitespaces?

Input

The input consists of several test cases. Each test case consists of a single line containing two numbers separated by a space N and K. The end of the test cases is given by the end of file (EOF).

- $1 \le N \le 10^6$
- $0 \le K \le 10^6$

Output

For each test case print in one line the number of different letters that can be written that has length N and exactly K whitespaces modulo $10^9 + 7$;

Example

Input	Output

3 0
3 1
10 3
	1
	4
	286

Problem E. Extended simulation

Source file name: `extended.c, extended.cpp, extended.java`
Input: `Standard`
Output: `Standard`
Author('s): `Juan Pablo Marín Rosas - CUCEI México`

Finally the simulation box has arrived, a simulation box has a number of N spots, a number M of links that link these spots, and a ball that will run on the simulation box during a simulation.

Linking on the spots is directional, this means, over a link something can pass only in one direction not both, if you want to do this, then the simulation box requires two different links i.e link $A- > B$ links the spot A with spot B allowing the ball to go from A to B but not from B to A.

The interesting part is running a simulation on the box, for this, the only thing required by human interaction is putting the ball in one of the spots. Once the ball is placed in the spot the ball will begin moving between the links, selecting a link at random that has as source the spot where the ball is currently placed and then moves to the destination of that link, the simulation finishes when the ball reaches the spot where it started.

There is one property in the box that we want to test, we know the ball moves at random but there may be some links that move the ball to one place where even if we run the simulation forever the ball will never reach its initial spot (simulation will be extended, nevec finishes), we call this links "lost links". Since having the simulation running forever won't help on determining wheter or not the ball will reach it's initial state, we ask you to write a program that helps us find given the simulation box configuration (spots and links) answer for each spot asked how many "lost links" can be reached if the simulation is started in that spot.

Input

The input consists of several test cases. The first line contains a single number T, the number of test cases to follow. Each of the T test cases starts with a line containing two integer numbers N and M. The next M lines contains two numbers A and B $(1 \leq A, B \leq N)$ that represents there exists a link from A to B. After the links description there will be a line with a single integer Q ($1 \leq Q \leq N$) the number of spots we are interested on knowing the number of lost links, followed by a line with Q integers each of these is a spot to answer how many "lost links" can be reached if the simulation is started in that spot.

- $1 \leq N \leq 1000$
- $1 \leq M \leq \frac{N(N-1)}{2}$
- $1 \leq Q \leq N$

Output

For each test case print exactly Q lines, the $i - th$ line will have the answer for the $i - th$ spot asked.

Example

Input	Output
1	1
7 8	1
1 3	0
3 2	
2 1	
1 4	
4 5	
5 6	
6 4	
4 7	
3	
1 6 7	

Explication

There will be only one test case. The test case has a box with 7 spots and 8 links. Then you are asked for 3 spots: 1, 6 and 7. From spot 1 and 6 only one "lost link" can be reached, from spot 7 there are no "list links" reachable.

Problem F. Friendly sum

Source file name:	`friendly.c, friendly.cpp, friendly.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

John was joking with his friends about how slow all of them sum. Then to improve their sum velocity all the friends decided to play a simple game. At the beggining each of the N friends pick a number, and will get a list of randomly selected friends each friend on the A list will be game peers, note that since each one receive a list, if B is in A list, not necessarily A is in B list.

After all this setup finishes, all the friends will run K rounds, each round consist on summing the numbers of all the game peers (i.e A will sum the number that each of the friends in his list have), all friends will wait for the others to finish summing, and then, when all of them finished, they change their number with the sum they got.

John doesn't like to play this kind of games, so he gave you the number of friends, the number each friend picked at the beginning, the list of each friends game peers and the number of rounds to run. He wants your help to determine what will be the number each friend has when all the K rounds have finished.

Input

The input consists of several test cases. Each test case begins with two numbers N and K. Followed by N lines, each of these lines starts with two numbers P_i which is the number the friend i selected at the beginning and L_i, which is the number of friends on the list of the $i-th$ friend, the rest of the line contains L_i numbers, each is one of the friends in i game peers list.

- $1 \leq N \leq 60$
- $1 \leq K \leq 10^9$

Output

For each test case print N lines. The $i-th$ line contains a single number, the number that the $i-th$ friend has after the K rounds were played. Since this number can be large print the result modulo 10^9+7.

Example

Input	Output
`3 2`	`40`
`10 2 2 3`	`40`
`10 2 1 3`	`40`
`10 2 1 2`	

Explication

There are 3 friends and will play 2 rounds.
Friend 1 picked the number 10 and his peers are the friends 2 and 3.
Friend 2 picked the number 10 and his peers are the friends 1 and 3.
Friend 3 picked the number 10 and his peers are the friends 1 and 2.
On the first round, friend 1 will sum $10 + 10 = 20$.
Friend 2 will sum $10 + 10 = 20$ and friend 3 will sum $10 + 10 = 20$. After all summed, they change their numbers and now Friend 1 has the number 20, friend 2 has the number 20 and friend 3 has the number 20.
On the second and last round, Friend 1 will sum $20 + 20 = 40$, friend 2 will sum $20 + 20 = 40$ and friend 3 will sum $20 + 20 = 40$. After all this they change their numbers and now Friend 1 has the number 40, friend 2 has the number 40 and friend 3 has the number 40.

Problem G. Gatuno's Fiber

Source file name:	`base32.c, base32.cpp, base32.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Félix Arreola - CUCEI México`

The students of Internet Programing are designing several new standards to improve the Internet as we know it. In fact, today have invented a new way of transmitting information thousand times faster than the optical fiber. The new standard name is Gatuno's Fiber.

Oddly, the medium has a small limitation, it can only transmit lowercase letters of the English alphabet (a-z) and the symbols ! @ # $ % & (exclmation mark, at sign, number sign, dollar sign, percent sign, ampersand sign)

One file can have a lot bytes outside the allowed letters (and signs), for this reason, the development team will also use a codification standard called Base32. In this codificaction schema, all bytes converted to binary. Next, are concatenated in a large string of bits. The bits are taken 5 by 5 to form a letter from 0 to 31, which corresponds to a symbol allowed in Gatuno's Fiber as show next:

Bits	Symbol	Bits	Symbol	Bits	Symbol	Bits	Symbol
00000	!	01000	c	10000	k	11000	s
00001	@	01001	d	10001	l	11001	t
00010	#	01010	e	10010	m	11010	u
00011	$	01011	f	10011	n	11011	v
00100	%	01100	g	10100	o	11100	w
00101	&	01101	h	10101	p	11101	x
00110	a	01110	i	10110	q	11110	y
00111	b	01111	j	10111	r	11111	z

For every 5 input bytes, they generate 8 output symbols. When there are not enough bytes to form the 8 symbols, is filled with zeros until a symbol is completed. For example, with 1 input byte, there are 2 output symbols (1 byte has 8 bits, so you need at least 2 symbols, 10 bits, to represent the byte).

Your task is help the students of Internet Programing to write a base32 encoder.

Input

The input consist of integer numbers $0 \leq N \leq 255$, one per line. Each integer represents a byte for encoding to Base32. The end of the bytes is given by the end of file (EOF).

Output

You must print the encoded symbols in base32 for the input bytes, printing a maximum of 80 symbols per line and followed by a newline. The last line must have a newline ending too.

Example

Input	Output
219 232 58 99 46	vjo$osti

Input	Output
132 101 58	klmno

Problem H. Hiding Sequence

Source file name:	`hiding.c, hiding.cpp, hiding.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

This problem is easy and fast to read, is it also easy and fast to solve ?

We call a hiding sequence a sequence that the sum of all its elements is equals to 0. As an example the sequence $(1, -1, 2, -2)$ is a hiding sequence but the sequence $(1, -1, 2)$ is not.

Given a list of N numbers, your task is to count how many sequences the list has that are hiding sequences. A sequence on the list can be obtained selecting two positions on the list and taking all the elements between them inclusive. (i.e $(1, -1, 2)$ is a valid sequence from the list $(1, -1, 2, -2)$, but $(1, 2)$ is not).

Input

The input file consists of two lines: The first line contains a single number N. The second line contains N integer numbers, each A_i from the list.

- $1 \leq N \leq 10^6$
- $-10^6 \leq A_i \leq 10^6$

Output

Print a single integer, the number of valid sequences in the list that are a hiding sequence.

Example

Input	Output

`5`
`1 -1 1 -1 1`
	`6`

Explication

The valid sequences that are hiding sequences from the list $1, -1, 1, -1, 1$ From 1st to 2nd element : $1, -1$
From 1st to 4th element : $1, -1, 1, -1$
From 2nd to 3rd element : $-1, 1$
From 2nd to 5th element : $-1, 1, -1, 1$
From 3rd to 4th element : $1, -1$
From 4th to 5th element : $-1, 1$

Problem I. Iterating Wheel

Source file name:	`iterating.c, iterating.cpp, iterating.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Juan Pablo Marín Rosas - CUCEI México`

The iterating wheels is a set of N wheels that will rotate when you push a button. During it's rotation the $i-th$ wheel will pick a token that is positioned just below it on the $i-th$ position and will move it to the P_i position.

It is warrantied that each position is reached only by one wheel and that all wheels reach only one position to move the token. At the beginning token 1 is below wheel 1, token 2 is below wheel 2 and so on.

Given the number of wheels in the iterating wheel, and the position P_i to which each wheel moves the token below it, your task is to determine what is the minimum number of times you have to press the button to leave the tokens in the same position where they started before the first time the button was pressed.

Input

The input file consists of several test cases, the first line for each test case starts with a single integer number N. Followed by one line containing N integers separated by one space, where the $i-th$ number is the value P_i. The end of the test cases is given by the end of file (EOF).

- $1 \leq N \leq 10^6$

Output

For each test case you must print a single number, the number of times you have to press the button to leave the tokens in the same posistion where they started. Since this number can be very large print it modulo $10^9 + 7$

Example

Input	Output

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

Explication

Tokens start in the order 1,2,3,4. The first time the button is pressed the tokens will be in the order : 2,1,4,3 The second time the button is pressed the tokens will be in the order : 1,2,3,4 Then, pressing two times the button the tokens will be in the start position.

Problem J. Jacksonville Police Departament

Source file name:	`jacksonville.c, jacksonville.cpp, jacksonville.java`
Input:	`Standard`
Output:	`Standard`
Author('s):	`Félix Arreola - CUCEI México`

Emergency!

The police in Jacksonville are following a suspect wanted for theft. In fact, he stole the ACM Contest's problem set. Lucky for the police, the thief hide inside a building. The building is surrounded and the police is about to enter.

But, the building has K departments, each identified by a number from 1 to K. After a quick search on the ACM Search Engine, the police found the name of suspect. He has an obsessive compulsive disorder and he only can enter on departments where the number is divisible by one of the thief favorite's numbers.

After another search on ACMBook profile page, they found that the thief has N favorite numbers. The police needs to know how many departments (in the worst case) is going to check.

Help the police to calculate this number in order to recover the lost problem set.

Input

The input consist of several test cases. Each test case consists of a line containing the numbers K and N. After that, there are N lines, each with one of the thief's favorite number F. The end of the test cases is given by 0 0

- $1 \leq K \leq 255$
- $1 \leq N \leq 20$
- $1 \leq F \leq 1000$

Output

For each test case, you should print the maximum number of departments that will be checked.

Example

Input	Output

16 2
3
5
100 1
2
20 3
2
3
4
0 0
	7
	50
	13