Problem L. Providers

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

There are N providers and M internet users in your town. Every user needs internet, but does not need more than one provider. Every user knows, what providers are available to him. Every provider can accept connections from not more than K_i users. You must find the maximal quantity of users, that can be connected to internet at the same time.

Input

The first line contains two integers N ($1 \le N \le 50$) and M ($1 \le M \le 500$). The second line contains N numbers K_i ($1 \le K_i \le 50$). Each line of next M lines contains a lists of providers, available for corresponding user - a set of nonrepeating numbers from 1 to N, separated by single spaces. List of providers is terminated by zero.

Output

Output one number - maximal quantity of users online.

Sample input and output

standard input or input.txt	standard output
2 5 3 2 1 2 0 1 2 0 1 0 1 0 2 0	5

Problem M. Primes

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Your task is simply to find the quantity of prime numbers in the range from a to b.

Input

The first line contains two integers a and b, separated by a space ($2 \le a \le b \le 10^{12}, (b - a) \le 10^7$).

Output

Output the quantity.

Sample input and output

standard input or input.txt	standard output
10 20	4

Problem N. Bishops

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

A bishop is a piece in the board game of chess. The bishop has no restrictions in distance for each move, but is limited to diagonal movement, forward and backward.

Consider a chessboard of size NxN.

Your must find the maximum number of bishops that can be placed on the chessboard, such that no two bishops can attack each other. Some cells are damaged. Bishops cannot be placed on these cells, but they can attack through them.

Input

The first line contains one integer $N (1 \le N \le 100)$ - size of the chessboard. The following N lines describe the chessboard. Each line contains N symbols. '.' is entire cell, '#' - damaged.

Output

Output a single integer - the number of bishops that can be placed on the chessboard under the given restrictions.

Sample input and output

standard input or input.txt	standard output

`2`
`..`
`..`
	`2`

`2`
`.#`
`#.`
	`1`

Problem O. Unusual lottery

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

In ordinary lotteries odds to win a prize are proportional to the number of bought lottery tickets. Organizers of a new lottery decided to encourage participants of the lottery to buy as much as possible lottery tickets by the fact that probability to win grows faster than the number of bought lottery tickets, namely proportional to the square of the number of the tickets bought by a participant.

There are three grand prizes in the lottery. Let's name them the first, the second and the third ones. Drawing of the lottery takes place in the following way. Stones are placed into a box. The number of stones corresponding to each participant is equal to the square of the number of the lottery tickets bought by that participant. Then stones are shuffled in the box and the winning stone is picked out of the box. The stone determines participant winning the first prize. Prior to the drawing of the second and the third prizes all stones of the first prize winner are removed from the box, so the first prize winner does not participate in the drawing of the next prizes.

After removing of the stones of the first winner, stones in the box are shuffled again, and again the winning stone is picked out of the box. The stone determines participant winning the second prize. Prior to the drawing of the last third prize all stones of the second prize winner are removed from the box, so the first and the second prize winners do not participate in the drawing of the third prize.

After removing of the stones of the second winner, stones in the box are shuffled the last time and a stone determining participant getting the third prize is picked out of the box. The drawing of the three grand prizes complete.

If happens that all stones are taken out of the box (it is possible when the number of participants of the lottery is less than 3) the remaining prizes are not assigned to any participant.

In this problem you are given the number of the lottery tickets bought by each of the lottery participant. You are to find the probability of winning the first, the second and the third prize for each of the lottery participants.

Let's consider the computation of the probability using the following example. There are 2 participants in the lottery, the participant A bought 2 lottery tickets and the participant B bought 1 lottery ticket.

For the first prize drawing the box will contain 2x2=4 stones of the participant A and 1x1=1 stone of the participant B, total 4+1=5 stones. The probability of winning the first prize for each participant is equal to the number of stones of that participant in the box divided by the total number of stones in the box. For the participant A the probability is equal to $4/5$ and for the participant B the probability is equal to $1/5$. Because of small number of the participants in this case the participant not winning the first prize certainly gets the second prize. The probability to win the second prize is $1/5$ for the participant A and is $4/5$ for the participant B.

Input

The first line of the input contains the number of the lottery participants N which is integer from 0 to 256 inclusive. Each of the following N lines contains the number of the lottery tickets boughts by a lottery participant which is integer from 0 to 256 inclusive.

Output

The output should contain N lines. One line describe one lottery participant. Each line should contain 3 fixed point numbers with exactly 3 digits after decimal point. Numbers should be separated by single spaces. These numbers are the probabilities to win the first, the second and the third prize for the participant.

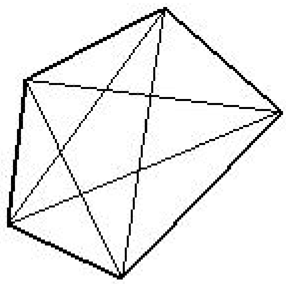
Sample input and output

standard input or input.txt	standard output
2 2 1	0.800 0.200 0.000 0.200 0.800 0.000

Problem P. Polygons

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

You are given a convex polygon with N vertices. Find the total number of parts it is divided by all possible diagonals, assuming, that no three of them intersect at one point.



Input

The first line of input contains one integer $N (3 \le N \le 50000)$.

Output

Output a single number - the number of parts.

Sample input and output

standard input or input.txt	standard output
3	1
4	4

Problem Q. Liars and knights

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

NxM people have arrived to the congress of the largest party in Byteland, and they were all seated in N rows, M people in one row. The party consists of two fractions - liars and knights. The representatives of the fraction of knights always speak the truth, and the liars always lie. After the congress each of the participants said that he had been sitting with the representatives of both fractions at his sides. Two participants are considered to be neighbors, if they are sitting next to each other in one row, or if they occupy the same places in two adjacent rows.

The chairman of the party does not know how many there were liars and knights. So he wants to understand with what minimal number of liars such a situation could take place.

Your task is to write a program that will find out the minimal number of liars and the corresponding plan of seating the congress participants.

Input

The first line of the input gives two integers N and M - the number of rows and seats in each row, correspondingly ($1 \leq N \leq 7$, $1 \leq M \leq 100$).

Output

Write in the first line of the output one number - the minimal quantity of liars in the required seating. The following N lines should show the seating itself, one row in each line. A knight must be shown by the symbol "." (point), and a liar by "x" (small Latin letter x). There should be no other symbols in the lines, not even the spaces. In case you find several ways of seating, show one of them.

Sample input and output

standard input or input.txt	standard output

`2 3`	`2`
	`..x`
	`x..`

`6 6`	`10`
	`x...x.`
	`..x...`
	`x....x`
	`...x..`
	`.x...x`
	`x..x..`

Problem R. Sequence

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

You are given an infinite sequence of integers:

$$A_0 = 0 \tag{1}$$

$$A_i = (A_{i-1} + q) \mod p, \text{for all } i > 0 \tag{2}$$

Where p is a prime number, $a \mod p$ is reminder of division a by p.

Your task is to count how many integers in the range [0..p-1] are not included in this sequence. Notice that p can be very large.

Input

The first line contains one prime integer p. The second line contains one integer q. ($0 \le q \le p \le 10^{100}$).

Output

Output a single number - the answer for the problem.

Sample input and output

standard input or input.txt	standard output
5 1	0

Problem S. Coins

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Monetary system in Byteland contains of N kinds of coins of different denominations. During ATM software developing, there occured a problem to give out the requested sum with minimal quantity of coins. To solve it, the following greedy algorithm was introduced. At first, maximal quantity of coins of greatest denomination are taken. Then, for the remaining sum, maximal quantity of coins of lesser denomination (next one in descending order) are taken, and so forth.

For example, we have coins of denominations 1, 2 and 5. To pay sum of 13, two coins of 5 are taken, then one coin of 2 and then one coin of 1 are taken.

Your task is to define, if this algorithm works correctly in all cases for a given monetary system.

Input

The first line of the input contains N ($1 \le N \le 1000$). The second line contains N coin denominations in ascending order, separated by single spaces ($1 \le D_i \le 1000$). The first denomination is always 1.

Output

If the introduced algorithm really gives out any sum with minimal quantity of coins, output 0. Otherwise output the minimal sum, that the algorithm will give out with non-minimal quantity of coins

Sample input and output

standard input or input.txt	standard output
3 1 2 5	0
4 1 3 4 5	7

Problem T. Galls village

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

As you know Caesar is a conqueror. He decided to occupy N villages in a very unusual way. Each day he chooses some nonempty subset of not conquered villages and tries to capture it. If he succeeds, those villages will be passed into his possession, else Caesar will set them free (freedom for all subset, but only for that day). This process repeats on the next day until there is at least one not captured village.

Any subset of villages can be chosen with equal probability. After choosing some subset, Caesar tries to capture villages one by one in any order.

Galls are living in the first village. Your task is to find the probability that someday all villages except it will be conquered.

Input

The first line contains one integer $N (1 \le N \le 15)$ - the number of villages. The following N lines contain integer probabilities P_i - probability of capturing i-th village in percents $(0 < P_i \le 100, 1 \le i \le N)$.

Output

Output a real number - the probability that someday only the first village will remain free. Answer should be accurate to 10^{-5}.

Sample input and output

standard input or input.txt	standard output
2 100 100	0.333333333
3 28 62 96	0.503901145

Hint:

Consider second example. At first day Caesar can choose any nonempty subset with equal probability:

(1,2,3), (1,2), (1,3), (2,3), (1), (2), (3)

If he chooses second subset, next day the third village will be the only free village with probability $0.28 \cdot 0.62$, else all villages will be left unoccupied

Problem U. String manipulations

Input file:	`standard input or input.txt`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

You are given a string S, consisting of lowercase Latin letters. The length of string is L. There are N small strings. You should concatenate all of them in any order to get string T. Your task is to count the number of different strings T, which are subsequences of string S.

Input

The first line contains L lowercase Latin letters ($1 \leq L \leq 10000$). The second line contains integer $N (1 \leq N \leq 9)$ - the number of small strings, which should be concatenated. The following N lines contain sequences of lowercase Latin letters. The total length of small strings does not exceed 1000 symbols.

Output

Output a single number - the answer for the problem.

Sample input and output

standard input or input.txt	standard output
abracadabra 2 aaa a	1
abracadabra 2 ara bra	2