



Problem A. Permugame

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.5 sec

Ksyusha got a great present for her birthday - a “Permugame”. Gear for “Permugame” consists of the two permutations of size N : P_i , Q_i and a square board $N \times N$ cells. According to the rules cells (i, j) and (a, b) are connected to each other by a tunnel if and only if either $(a = i \text{ and } b = Q_j)$ or $(a = P_i \text{ and } b = j)$. The only goal of the game is to find such a minimal integer k ($k > 0$), that every cell of the board can be painted in one of the k colors. The only restriction is that any two connected cells should have different colors. Ksyusha is too lazy to play “Permugame” so she asked you to figure out the answer.

Input

In the first line you are given a single integer N ($1 \leq N \leq 100000$) - size of the board and permutations P and Q . In the next line permutation P_i ($1 \leq P_i \leq N$) is given as a list of integers separated by spaces. In the third line permutation Q_i ($1 \leq Q_i \leq N$) is given in the same format. It is guaranteed that $P_i \neq i$ and $Q_i \neq i$ for each i in $[1, N]$.

Output

Print the minimal integer k ($k > 0$) such that every cell of the $N \times N$ board can be painted in one of the k colors.

Sample input	Sample output
2 2 1 2 1	2
4 2 4 1 3 3 4 2 1	2
5 5 3 4 2 1 2 5 4 3 1	3



Problem B. Pluses and asterisks – 2

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 1 sec

Given a sequence of integers $a_1 ? a_2 ? a_3 ? \dots ? a_N$ one may replace each $?$ with either $+$ or $*$. After that, the expression is evaluated according to arithmetic rules, where $+$ denotes addition, $*$ denotes multiplication. Multiplication's precedence is higher than addition's, i.e. $2+2*2$ is 6, not 8. In how many ways it is possible to make result equal to R ?

Input

The 1st line contains space-separated integers N and R , denoting quantity of values a_i and required result respectively. The 2nd line contains space-separated values $a_1, a_2, a_3, \dots, a_N$. $3 \leq N \leq 38, 1 \leq R \leq 2^{42}, 1 \leq a_k \leq 2^{17}$.

Output

Your program should write exactly one integer — the quantity of ways to obtain exactly R .

Sample input	Sample output
4 8 2 2 2 2	5

Note: The five ways are: $2+2+2+2$; $2+2+2*2$; $2+2*2+2$; $2*2+2+2$; $2*2+2*2$.



Problem C. Sleeping cars

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 0.5 sec

A train contains sleeping cars, marked with letter k , and sitting cars, marked with letter p . You need to find the biggest number of sleeping cars that follow each other in the train.

Input

The only line at input contains a sequence of letters k and p , with length from 1 to 1000 symbols.

Output

You need to print one number – maximum number of sleeping cars that follow each other.

Sample input	Sample output
kpkkp	2



Problem D. Numbairs

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.1 sec

Consider number of the form $a^{a^{\dots^a}}$ where a is a positive integer that can appear in the notation twice or more. Let us call such numbers *numbairs* (which stands for *number + stairs*). For instance, both $27 = 3^3$ and $16 = 2^{2^2}$ are numbairs. Number 1 is, too, a numbair since $1 = 1^1$. Find out how many numbairs there are between 1 and a number n given to you (inclusive).

Input

The input contains single integer n , $1 \leq n \leq 10^9$.

Output

Output number of numbairs not exceeding n .

Sample input	Sample output
5	2



Problem E. Memory Manager

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.5 sec

You are working on the compiler of your own brand new programming language and you have already implemented everything except for memory manager (MM). MM is responsible for allocating objects in memory and deleting them when they become unnecessary. In total, there are N consecutive bytes of memory in your computer. Each object requires certain amount of consecutive bytes in memory. During its execution, the program can send queries of two types to MM:

- 1) Allocate an object of size A bytes. Note that this size can be different for different objects. Two objects cannot have shared memory.
- 2) Remove the given object, which was previously allocated.

You know *in advance* all M queries, which will be sent to MM. Find the way to process them all or report that it is impossible. *It is guaranteed that there will be not more than two queries of the second kind.*

Input

The first line of the input contains two integers N and M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$). Each of the next M lines contains two integers a and b . If $a = 1$, then b is equal to the size of an object you need to allocate. If $a = 2$, then b is equal to the index of query which was already processed and means that you should remove an object allocated in that query. It's guaranteed that the input is correct, e.g. you'll never be asked to remove an already removed object, etc.

Output

If it is impossible to perform all queries, print one line with the word "IMPOSSIBLE" (without quotes).

Otherwise, for each query of the first type (the ones with $a = 1$) you should output one number in its own line, which represents the position of the first byte of an allocated object.



Bytes in memory are numbered from 0. If there are several valid answer sequences, you may output any of them.

Sample input	Sample output
10 5	5
1 5	0
1 5	0
2 2	
1 5	
2 1	



Problem F. Construction

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 0.5 sec

After finishing his chalet construction, Stepan still has N wooden boards, with lengths L_1, \dots, L_N . He decided to build a fishing deck on a lake using these boards.

Stepan believes that the longer deck is, the more fish he will catch!

Moreover, Stepan, as all fishermen, is very superstitious and trusts all omens. One of it – the deck can be build using whole boards only (boards can be cut but not joined). Now Stepan is interested what maximum deck length D can be obtained, if needs to use L boards.

Input

The input contains integers N, M, L_i ($1 \leq N \leq 10000, 1 \leq M, L_i \leq 2 \times 10^9$) – number of available boards, number of boards to be used to build the deck, lengths of available boards.

Output

You need to print one integer D – maximum possible deck length or 0 (zero).

Sample input	Sample output
4 4 5 5 3 6	3



Problem G. Divisible by 3

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 0.5 sec

Let's consider a sequence 1, 12, 123, 1234, 12345, 123456, 1234567, 12345678, 123456789, 12345678910, 1234567891011,

Write a program that determines how many elements of this sequence among first N are divisible by three.

Input

The only line on the input contains one positive integer N ($1 \leq N \leq 2^{31}-1$).

Output

You need to print one number you've found.

Sample input	Sample output
4	2



Problem H. Number Set

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.1 sec

You are given a set of n positive integers. Find its largest subset A such that every number in A is a divisor of the product of all the other numbers in A .

Input

The first line of the input contains single integer n , $3 \leq n \leq 10^4$. The second line contains n positive integers in strictly ascending order. None of these numbers exceeds 10^6 .

Output

The first line of the output should give the size of the largest subset A satisfying the condition. In the second line, you should list all the numbers in this subset in ascending order. If there is more than one subset satisfying the condition, output any one of them. You can rely on that it will be at least one such subset with at least three numbers.

Sample input	Sample output
8 2 3 4 5 7 8 9 10	5 2 4 5 8 10



Problem I. Counting-Out

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.5 sec

N children, standing in a circle, are using counting-outs to determine who will leave the circle. It means, they recite counting-out rhymes, pointing each rhyme's word on next (by circle) child. The child being pointed on at the last word of the rhyme leaves the circle. The process repeats K ($1 \leq K \leq N$) times, restarting from the child following the one who just had left the circle. Counting starts from lower numbers to higher, but counting-out rhymes are different.

Write a program which reads number of children, number of counting-out rhymes, and numbers of words in used counting-out rhymes, and finds the order in which children will leave the circle.

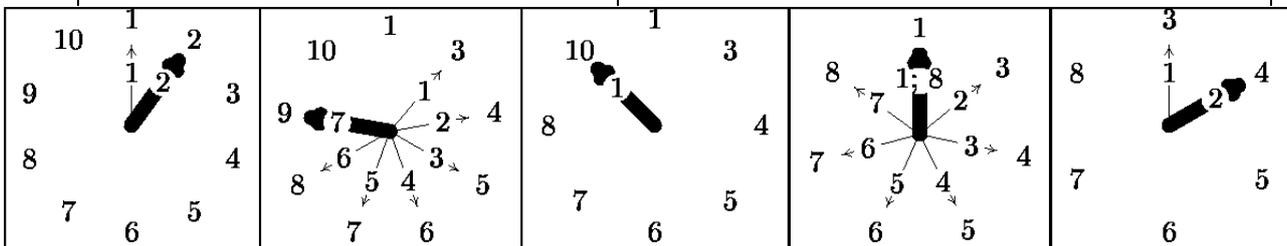
Input

The 1st line contains space separated number of children N ($1 \leq N \leq 10^{18}$) and number of used counting-out rhymes K ($1 \leq K \leq \min(N, 10^5)$). The 2nd line contains K integers (each in range $1 \leq a_i \leq 10^{18}$) — numbers of words in the counting-out rhymes, used the 1st, the 2nd, ..., the K -th time.

Output

The only line of the output should contain space-separated sequence of K numbers — the numbers of the children leaving the circle after the corresponding counting-out. Initially, children are numbered from 1 to K , and the 1st is the child who is pointed on when reciting the 1st word of the 1st counting-out rhyme. Further, each child keeps the same number, irrespectively of how children leave the circle.

Sample input	Sample output
10 5 2 7 1 8 2	2 9 10 1 4





Problem J. Fence

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 0.5 sec

Emos have relocated to newly built houses and decided to build a fence around their settlement. Emos are strange people, and they will cry if at least one section of the fence is not parallel to coordinate axis. Therefore, you have a task – to build a fence of minimal length around Emo's settlement in a way that all houses are located inside the area limited by the fence. The fence should be a polygon without self-intersections and self-tangencies with sides parallel to coordinate axis. Houses are points having coordinates. Some houses are allowed to be located at the fence itself.

Input

The first line at input contains integer N ($2 \leq N \leq 100\,000$) – number of houses. The following N lines contain descriptions of houses – each line contains two integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$). All houses are located at different points, at least two x -coordinates and y -coordinates are different.

Output

You need to print one number with at least six digits after decimal point – minimal possible length of the fence.

Sample input	Sample output
4 1 0 0 1 1 1 0 0	4.000000



Problem K. Dossiers

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time limit: 0.5 sec

The following information about all pupils of a school is available: surname, first name, class, date of birth. Depending on situation, Director may need to sort it in different modes. Full list of standard sort modes is:

surname — by surname, ignoring first name;

fullname — first by surname, then by first name for equal surnames;

birthyear — by year of birth, ignoring date inside year, from earlier years (elder pupils) to later years (younger pupils);

birthdate — by date of birth, i. e. first by year, then by month, then by day, from earlier dates (elder pupils) to later dates (younger pupils);

birthday — by birthdays, from 01.01 to 31.12 (in DD.MM format), in calendar order, considering the same dates of different years equal;

grade — by class (1–11) as year of study, ignoring letter;

class — first by grade (year of study), then by class letter (A–Z) for equal grades.

Sometimes Director needs to use first one of the modes, then another (for pupils equal by first mode), etc. For example, “birthyear grade surname” means to place firstly all pupils who were born in 1998, then all who were born in 1999, etc; pupils, who were born in the same year, should be sorted by grade (class ignoring letter); finally, pupils who were born in the same year and are studying in the same grade should be sorted by surnames.

Input



The 1st line contains required sort mode, as one or some of words surname, fullname, birthyear, birthdate, birthday, grade, class. At least one word will appear; no word can appear more than once. If there are several words, they are separated by single spaces.

The 2nd line contains the quantity of pupils N ($2 \leq N \leq 12345$).

The following text contains N four-line groups, describing a pupil. First line of each group contains surname. Surname must begin with capital Latin letter, which can be followed by sequence of Latin letters, hyphens (-) and apostrophes ('). Second line of each group contains first name. First name must begin with capital Latin letter, which can be followed by sequence of Latin letters, hyphens (-), apostrophes (') and spaces (). Third line of each group contains class — integer number from 1 to 11 and exactly one capital Latin letter, without space between. Fourth line of each group contains date of birth, given in format DD.MM.YY, i. e. exactly two digits for day-of-month, point, exactly two digits for month, point, exactly two digits for year

All dates of birth are correct (for example, it cannot be 30.02) and are in range from 1990 to 2030. It means that, for example, $97 < 02$, because 97 means 1997 and 02 means 2002. Typically, pupils of higher classes have earlier dates of birth, but this rule can have exceptions.

Major (at least 99.9%) of surnames and first names are not longer than 20 characters, but all we know about the rest 0.1% is that their total length is not greater than one million characters. Surnames and first names may end with Latin letter or apostrophe, and cannot end with hyphen or with space. You should compare surnames with surnames and first names with first names, using standard case-sensitive string comparing, implemented in programming language (even if names contain apostrophes or other non-letter symbols).

Output

Program should print exactly N lines. Each line should contain information about one pupil, in format: class, comma, surname, comma, first name, comma, date of birth. Each of class, surname, first name and date of birth should be byte-in-byte the same, as in input data. Lines should be sorted according to the specified sort modes.



In all cases, when applying of the specified sorting modes to the specified input data leaves the possibility of different correct answers, program should output any one of them. For example, for specified sort modes “birthyear grade surname”, if there are pupils of the same year of birth, studying in the same grade and having the same surname — they can be written in any order, each exactly once.

Sample input	Sample output
birthyear grade surname	11A, Ivanov, Kirill, 12.01.97
8	11B, Ivanov, Andrey, 23.12.97
Gonsales	11B, Petrova, Natalia, 23.08.97
Anna Maria	9A, Smirnov-Kovalenko, Egor, 13.11.98
5A	10A, Shevchenko, Denis, 07.05.98
01.05.03	11B, Andreeva, Ekaterina, 07.02.98
Ivanov	5A, Gonsales, Anna Maria, 01.05.03
Kirill	3D, Ivanov, Ivan, 20.01.05
11A	
12.01.97	
Ivanov	
Ivan	
3D	
20.01.05	
Ivanov	
Andrey	
11B	
23.12.97	
Petrova	
Natalia	
11B	
23.08.97	
Andreeva	
Ekaterina	
11B	
07.02.98	
Shevchenko	
Denis	
10A	
07.05.98	
Smirnov-Kovalenko	
Egor	
9A	
13.11.98	



Problem L. Saucepans and lids

Input: standard input
Output: standard output
Memory limit: 256 Mb
Time Limit: 0.5 sec

Great disaster happened this morning in the cafe where you used to have your snack in university. Larysa Ivanivna, the cleaner, while sweeping the floor dropped one of cupboards and all kitchenware stored there has scattered over the floor. Luckily, this cupboard contained only saucepans with lids. However, some items bent or broke, so has been disposed.

Now the schoolmaster wants to calculate losses and understand how many new saucepans and lids should be bought. First, she needs to find out how many remaining saucepans can be covered with remaining lids.

Saucepans and lids are round shaped. A lid can cover a saucepan if and only if radius of the lid is not less than radius of the saucepan.

Input

The first line at input contains integers n, m ($1 \leq n, m \leq 1000$) – numbers of remaining saucepans and lids. The second line contains n integers a_i ($1 \leq a_i \leq 1000$) – radiuses of remaining saucepans. The third line contains m integers b_i ($1 \leq b_i \leq 1000$) – radiuses of remaining lids.

Output

The only line at output should contain one integer – maximum number of saucepots that can be covered by available lids.

Sample input	Sample output
5 5 4 8 1 2 5 7 2 4 6 5	4