Problem A. String merging

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In this problem you will be given two strings - A and B.

Your task is to find a string C such that it contains both A and B as substring and it will be shortest among all possible strings.

A substring of a string is a contiguous subsequence of that string. So, string **kbtu** is substring of string **kbtu open**, but string **fall** is not.

Input

The first line of the input will contain string A ($1 \leq |A| \leq 10^5$).

The second line of the input will contains string B ($1 \leq |B| \leq 10^5$).

It is guaranteed that both strings will contain only lowercase Latin letters.

Output

Print one string - C

Examples

standard input	standard output
compressing single	compressingle
can you	youcan
compressiondoneright doner	compressiondoneright

Problem B. Pretty necklace

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Temirulan wants to make a necklace as a present to his beloved girl. A necklace is a cyclic sequence of blue and red colored beads.

Temirulan already has one that consists of N beads. He knows that his girlfriend prefers red color over blue, so he decided to cut some subsequence of at least K beads from the original necklace such that the ratio between red ones and the number of chosen beads will be maximized.

Can you help him to find this maximum ratio?

Input

The first line contains two integers N, K ($1 \leq K \leq N \leq 5 \cdot 10^5$) — the number of beads on the thread and lower-bound of beads for the new necklace.

The second line contains the sequence of N integers separated by a single space a_i ($0 \leq a_i \leq 1$) - description of the original necklace.

$a_i = 0$ corresponds to blue and $a_i = 1$ corresponds to red color.

Output

Output the maximum ratio of red in the new necklace that could be cut from the original one. Assume two values are equal if they differ by 10^{-6} or less.

Examples

standard input	standard output
8 4 11101110	0.857142448425
8 4 11011001	0.833333015442
10 4 1001001001	0.599999427795

Problem C. Boring GCD

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You are given an array $a_1, a_2, a_3..a_n$. There are 4 types of operations with it:

- 1 l r x, for each $i \in [l, r]$ replace a_i with the value of x
- 2 l r x, for each $i \in [l, r]$ replace a_i with the value of $gcd(a_i, x)$ function
- 3 l r, output the value of $\max_{i \in [l,r]} a_i$
- 4 l r, output the value of $\sum_{i \in [l,r]} a_i$

Greatest common divisor $gcd(a, b)$ of two positive integers a and b is equal to the biggest integer d such that both integers a and b are divisible by d.

Input

The first line contains two integers n, q ($1 \leq n, m \leq 10^5$) — the number of array elements and the number of queries.

The second line contains n positive integers $a_1, a_2, .., a_n$ - initial state of the array.

Next m lines contain the description of the queries, one per line. Queries are formatted the same way as in the problem statement above.

It is guaranteed that $1 \leq l \leq r \leq n$ and $1 \leq x \leq 10^9$

Output

For each 3rd and 4th query type output answer for this query in a separate line.

Example

standard input	standard output
5 11 1 6 8 7 3 3 1 5 4 1 5 2 1 5 6 3 1 5 4 2 4 1 1 5 10 3 1 4 4 3 5 2 3 4 3 3 2 3 4 4 5	8 25 6 9 10 30 10 11

Problem D. Students love

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Nurdaulet and Zharaskhan are coaching students. To each student they have their own attitudes, it can be expressed as number a_i (for Nurdaulet) and b_i (for Zharaskan) that are called `live index` of students. Askar asked them to calculate the `unfair attitude rate`. `Unfair attitude rate` is the difference between the largest and the smallest `love index`. In order to not show their possibly large `unfair attitude rates`, they decided to cheat: each shuffle his array, then form new array $c_i = a_i + b_i$ and show the rate of new formed array to Askar. What is the minimal possible rate they can achieve?

Input

On the first line you are given single integer $n(1 \leq n \leq 200000)$. On the second line you are given n integers $a_i(-10^6 \leq a_i \leq 10^6)$. On the third line you are given n integers $b_i(-10^6 \leq b_i \leq 10^6)$.

Output

Output single integer, the answer to the problem

Example

standard input	standard output
2 -3 -5 3 5	0

Problem E. Excalibur

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

Excalibur is the legendary sword of King Arthur, sometimes also attributed with magical powers or associated with the rightful sovereignty of Britain. In order to be the King of the Great Britain, Alan decided to craft Excalibur. There is some really old magic, that allows you to create the sword. There are m words in the magic spell, each word is represented by some integer w_i. In order to cast the spell, you need to construct connected graph with n vertices and m edges without loops or multiple edges. Edge weights should be weights from the magic spell. In other words, weights of this graph should form permutation of w_i. In this graph the sum of the edges of minimum spanning tree should be equal to X. Minimum spanning tree is the undirected connected graph with $n-1$ edges without cycles, such that sum of the edges in this tree is minimum possible.

Input

On the first line you are given 3 integers $n, m, X (2 \le n \le 100, 0 \le m \le 1000, 0 \le X \le 4000)$ In the second line you are given m integers $w_i (1 \le w_i \le 40)$.

Output

If it is not possible to craft Excalibur, on the single line output -1. Otherwise, on the first line output m. On the next m lines output 3 integers on each, $u, v, c (1 \le u \le n, 1 \le v \le n, u \ne v, 1 \le c \le 40)$ – edges of the graph. Weights should be some permutation of w_i from the input

Example

standard input	standard output
3 2 2 1 1	2 1 2 1 2 3 1

Problem F. GeoGame

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Alimzhan play interesting games a lot. One of them is `GeoGame`. In this game you stay at the plane, and your position can be described as point (x, y) in the plane. There are enemies with special powers: mages, warriors and tricksters. Their aim to confuse Alimzhan about his position, so that he won't know where he is. Each of them can affect Alimzhan's position in some way:
1. x y - warrior knocks and pushes by x towards OX axis, and by y towards OY axis.
2. a b - mage in point (0,0) rotate you around his position on $\frac{a*\pi}{b}$ radians counterclockwise.
3. b x y - trickster sends you in mirror world: he creates infinite mirror (some line), so that Alimzhan turns into his mirror image.
There are n actions your enemies can possibly do: $A_1, A_2, ..., A_n$. Alimzhan tired of games, so he asks your to answer q queries: what is his new position after consequent effects $A_l, ..., A_r$(in this order), if his starting position is (x, y)?

Input

In the first line you are given n ($1 \le n \le 3 \cdot 10^5$). In the next n lines you are given n actions - each action starts with t - type of operations:
1 x y - warrior's action ($|x| \le 100, |y| \le 100$)
2 a b - mage's action ($|a| \le b \le 42$)
3 b x y - trickster's action - mirror(line) is defined with 2 points $(b, 0)$ and (x, y). ($|b|, |x|, |y| \le 100$)
In the $(n + 1)$th line you are given number of queries q ($1 \le q \le 3 \cdot 10^5$).

In the new q lines you are given queries.Each query is 4 integers in format l r x y ($1 \le l \le r \le n$)

Output

For each of q queries output the final position as two real numbers. Your answer is considered correct if your absolute or relative error don't exceed 10^4.

Example

standard input	standard output
3 1 -1 2 2 1 2 3 -1 -3 2 3 1 1 5 0 1 2 5 0 1 3 5 0	4.000000 2.000000 -2.000000 4.000000 -5.000000 1.000000

Problem G. Graphity

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given weighted unordered graph with n vertices and m edges. Graph doesn't contain loops and can be drawn on the plane. NurlashKO has restored the graph on the plane, and build the following graph: the faces (regions) on the plane became vertices, and edges between them were the edges, that is common to these faces, e.g. if 2 faces share the edge in initial graph - add the edge between faces in new graph. Now NurlashKO wants to find the minimum spanning tree of the new graph. Minimum spanning tree is a connected tree with n vertices with minimal sum of the edges in the tree.

Input

In the first line you are given 2 integers n and $m(3 \leq n \leq 1000000, 0 \leq m \leq 200000)$ Next m lines contain 3 integers each: u, v, c, where u and v are the vertices, and c is the weight of the edge connecting these vertices. $(1 \leq u, v \leq n, u \neq v, -10^6 \leq c \leq 10^6)$

Output

Print single integer, the total weight of the minimum spanning tree.

Example

standard input	standard output
3 3 1 2 1 2 3 1 1 3 1	1

Problem H. Kill all termites

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are termites on the tree. Your task is to kill them all. The tree is an undirected connected graph with n vertices and $n-1$ edges. In order to kill termites you are able to poison some vertices. If the termite hits the vertex with poison, then he immediately dies. You don't know where termites are located initially. But you know that termites go to the random adjacent vertex each time, but if termite has passed edge (u, v), next edge should be different from (v, u), except the case, when termite hits the leaf (in this case, termite turns around and goes back). You need to poison minimum number of vertices, such that termites hits poisoned vertices after finite number of steps.

Input

On the first line you are given single integer n ($1 \le n \le 100000$). On the next line you are given $n-1$ integers p_i ($2 \le i \le n$), so there is an edge connecting p_i and i.

Output

On the single line output single integer – minimum number of poisoned vertices.

Examples

standard input	standard output
1	1
2 1	1
8 1 1 2 1 2 3 2	2

Problem I. XOR

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

This is an interactive problem.

Jury has hidden a binary sequence of length n. You know only the length n.

You should divide this binary sequence into two subsets with equal number of ones.

You can make queries of the following three types:

- 1 i ($1 \le i \le n$) — xor the i-th bit (You can make up to 777 queries for this type)
- 2 — ask number of ones in the sequence (You can ask only once)
- 3 — print the answer (Look at the output section)

Input

The first line contains single integer n ($1 \le n \le 777$) — the length of the hidden binary sequence. You should read this integer first.

Output

When your program is ready to print the answer, print two lines.

In the first line print 3.

In the second line print n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 2$) — $a_i = 1$ if i-th bit in the first subset, $a_i = 2$ if in the second subset.

Your program should terminate after printing the answer.

Interaction Protocol

To make a query of the first type, print 1 i ($1 \le i \le n$), where I is the index of element you want to xor.

After query of type 2 read single integers $value$ ($0 \le value \le n$), number of ones in binary sequence.

Your solution will get *"Idleness Limit Exceeded"*, if you don't print anything or forget to flush the output, including for the final answer .

To flush you can use (just after printing line break):

- fflush(stdout) in C++;
- System.out.flush() in Java;
- stdout.flush() in Python;
- flush(output) in Pascal;
- For other languages just google.

Example

standard input	standard output
4 4	 1 1 2 3 1 1 2 2

Note

In the example, initial binary sequence was

0 1 1 1

After query of type 1 it changed to

1 1 1 1

So, for the query of type 2 answer is 4

Problem J. LOL

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

10 hours before contest...

```
HAI 1.2

I HAS A PEN, GIMMEH PEN, PEN R MAEK PEN A NUMBR
I HAS A APPLE, GIMMEH APPLE, APPLE IS NOW A NUMBR

I HAS A LOL ITZ 100500
I HAS A TRASH ITZ 0
IM IN YR DREAM UPPIN YR LOL WILE DIFFRINT APPLE AN SMALLR OF APPLE AN
TRASH
  IM IN YR HOUSE UPPIN YR LOL WILE BOTH SAEM PEN AN BIGGR OF PEN AN APPLE
    PEN R DIFF OF PEN AN APPLE
  IM OUTTA YR HOUSE
  I HAS A PINEAPPLE ITZ PEN
  PEN R APPLE
  APPLE R PINEAPPLE
IM OUTTA YR DREAM
VISIBLE SUM OF PEN AN APPLE

KTHXBYE

BTW RLY LOL :D
```

Input

two cows

Output

happy life

Example

standard input	standard output
2 2	2

Problem K. woogle.we

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Did you know that in autumn people work better? Just because ancients used to harvest each autumn.

Temirulan decided to travel around Woogle offices. There are some two-way paths between some of the offices. Each of those paths costs some amount of wollars. We can assume that Temirulan has an unlimited amount of wollars. Temirulan pays for the path using his Waspi card. *Note that Temirulan can travel between all pairs of offices, and there is no self connecting paths.* Aidana worries about Temirulan. She wants Temirulan to spend less money on this travel. She wants to set a minimum possible limit on his card(Who understands these girls?), so that Temirulan won't see the difference. It means, that if Temirlan goes from office A to office B in the cheapest way (he always goes in an optimal way, ACM habit), then **each** his payment during travel won't be declined due to limit, for any office A and B.

Help Aidana to choose the minimum amount of wollars she can set on his card as limit and at the same time Temirulan would be able to travel from any office to any office by optimal way.

Input

In the first line of input given two non-negative integer numbers $n, m (2 \le n \le 100, 1 \le m \le n*(n-1)/2)$ — the number of offices, and the number of paths between them.

In the next m lines of input given by three non-negative integer numbers $v, u, w (1 \le v, u \le n, 1 \le w \le 10^5)$ — description of offices and cost of the path.

Output

Output one integer — the minimal limit for Temirulan's card.

Example

standard input	standard output
5 5 1 2 1 2 3 1 3 4 1 4 5 1 1 5 2	2

Problem L. Abusing 1Fit limits

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

1Fit — is a subscription-based startup, within 1Fit you can visit multiple gyms without buying separate subscriptions.

In 1Fit we have n fitnesses, m trainings and k sport types. All entities numbered consecutively starting from 1. Each training has $fitnessID$ and $sporttypeID$ specified.

There are r rules, each rule describes the book limit per user. In each rule specified fitness, list of sport types and limit, $limit = -1$ stands for an unlimited book, e.g. if $fitnessID = 3$ and $sporttypes = [1, 2, 6]$ and $limit = 7$ means that the number of books with 3rd fitness and sporttype equals to 1 or 2 or 6 cannot exceed 7. One training cannot be booked multiple times. It's guaranteed that each pair $fitnessID$ and $sporttypeID$ can appear in at most **one** rule as well.

Aisultan recently bought a 1Fit subscription to get more healthy. He sent q consecutive book queries. For each query server must return either 'yes' or 'no', is training booked successfully or the booking was incorrect. Write a program to process all Aisultan's queries.

P.S: Good luck on the contest, and special for you there's promo code: FALL19 which gives 10% discount on everything :)

Input

The first line of input given three positive integer numbers n, m and k ($1 \le n, m, k \le 100$) — the number of fitnesses, trainings and sport types, respectively. Each of the next m lines contains two integers $fitnessID$, $sporttypeID$ ($1 \le fitnessID \le n$, $1 \le sporttypeID \le k$) — description of trainings: fitness ID and sport type ID of training. Next line contains one positive integer r ($1 \le r \le 100$) — number of rules. Each of the next r lines contains the first ID of fitness, following by the number of sport types in this rule and list of sport type ids and limit at the end. The next line contains one positive number q ($1 \le q \le 100$) — the number of Aisultan's queries. Each of the next q lines contains one integer — description of query: ID of training.

Output

For each query output either 'yes' or 'no'.

Example

standard input	standard output
3 8 3 1 1 1 2 2 2 2 3 3 1 3 3 3 1 1 1 4 1 1 2 -1 2 2 2 3 2 3 1 1 -1 1 1 1 1 9 1 2 3 2 4 5 8 6 7	yes yes yes no yes yes no no yes

Note

In the example, we have 3 fitnesses, 8 trainings and 3 sport types. Training lists are: $(1, 1)$, $(1, 2)$, ..., i.e. training number 1 in fitness with ID 1 with sport type ID 1, training number 2 in fitness with ID 1 with sport type ID 2, ...

Have 4 rules:

1. $fitnessID = 1$ $sporttypeIDs = [2]$ $limit = -1$ — Aisultan can book trainings with $fitnessID = 1$ and $sporttypeID = 2$ unlimited times
2. $fitnessID = 2$ $sporttypeIDs = [2, 3]$ $limit = 2$ — Aisultan can book trainings with $fitnessID = 2$ and $sporttypeID = 2$ or $fitnessID = 2$ and $sporttypeID = 3$ no more than twice in total
3. $fitnessID = 3$ $sporttypeIDs = [1]$ $limit = -1$ — Aisultan can book trainings with $fitnessID = 3$ and $sporttypeID = 1$
4. $fitnessID = 1$ $sporttypeIDs = [1]$ $limit = 1$ — Aisultan can book trainings with $fitnessID = 1$ and $sporttypeID = 1$ no more than once

The first three queries Aisultan can book successfully. 4th query will be 'no' because Aisultan cannot book 2nd training twice. Also 7th query's response is 'no', because Aisultan cannot book training with $fitnessID = 1$ and $sporttypeID = 1$ twice.