Problem A. House Pricing Prediction

Input file:	darecosystem.in
Output file:	darecosystem.out
Time limit:	1 second
Memory limit:	256 megabytes

Samal is planning to sell her house, but she doesn't know what price to put. Fortunately, there is a history of N sold houses in this year, we call it *Training set*, all houses indexed from 1 up to N, so that she decided to find the most similar houses that have been sold and take the average of their prices. Each house has D features, i.e. area, room count etc. And for each house each feature represented as integer value. Having an engineering mind, she takes the following approach:

1. She computes an *Euclidian distance* between her house and each house in the Training set.
2. She chooses K houses from Training set with minimal *Euclidian distance*. If there is two or more houses has equal *Euclidian distance* then at first she choose house with maximum index.
3. After this she calculate average value of prices among those K houses. That's value declared as price prediction for her house.

For given information about her house and all Training set with all features calculate prediction of her house. See notes for some explanations.

Input

First line of input contains three positive integer numbers N, D, K ($1 \le N, D \le 500$, $1 \le K \le N$) — number of houses in Training set, number of features in each houses, value for prediction. Second line contains D integer numbers a_i — integer value of i-th feature. Next line N lines contains $D + 1$ integer numbers, in i-th of them $(b_{i,1}, b_{i,2}, ..., b_{i,D}, price_i)$ — $b_{i,j}$ is integer number of j-th feature of i-th house and $price_i$ price of i-th house. All integers in house characteristics are in range $[0, 100]$.

Output

Output one number — average house price among K nearest houses. Your answer will be considered correct it its absolute or relative error does not exceed 10^{-2}.

Example

darecosystem.in	darecosystem.out
4 2 3 1 2 0 0 23 4 1 37 2 5 43 3 2 93	53.000

Note

Euclidian distance between two houses is square root of sum of squares of differences in features, i.e. A is house with features $(2, 3, 1, 4)$ and B is a house with features $(1, 5, 4, 4)$, then *Euclidian distance(A, B)* = $\sqrt{(2-1)^2 + (3-5)^2 + (1-4)^2 + (4-4)^2} = \sqrt{1+4+9+0} \approx 3.74165739$.

Problem B. From XOR with love[again 2]

Input file:	xorlove.in
Output file:	xorlove.out
Time limit:	2 seconds
Memory limit:	32 megabytes

Again XOR, again query, again tree — романтика (c) Ануар Сериков

Where XOR there is no place for words. So let's move on with problem.

Given a rooted tree with n vertices, all vertices numbered from 1 up to n. Each vertex of the tree has a single number written on it, value of vertex. The root of the tree is vertex 1.

In this problem you have to process q queries. Each query is one the following 2 types:

- 1 v x — change all numbers written on the vertices of the subtree of the vertex v, i.e. if the number written on some vertex is y, substitute it with $x \oplus y$.
- 2 u v — print the minimum of the numbers written on the vertices of the simple path from the vertex u to the vertex v, i.e. more formally let's numbers on path from vertex u to vertex v will be $a_1, a_2, ..., a_k$ then answer to this query is minimum among this sequence.

Input

The first line contains two integers n ($1 \le n \le 3 * 10^4$) — the number of vertices in the tree, and q ($1 \le q \le 3 * 10^4$) — the number of queries.

The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{20}$) — the numbers written on vertices.

Then follow $n - 1$ lines, describing the tree edges. Each line contains a pair of integers x_i, y_i ($1 \le x_i, y_i \le n, x_i \ne y_i$) — the vertices connected by an edge, it's guaranteed that graph is connected tree.

The last q lines describe the queries. The first number on i-th line is $type_i$ — the type of the i-th query. If $type_i = 1$, then follows two integer numbers v_i and x_i ($1 \le v_i \le n$, $0 \le x < 2^{20}$), otherwise it follows with two integer numbers u_i and v_i ($1 \le u_i, v_i \le n$). All numbers in input are integers.

Output

For each query of 2nd type print one integer — answer to the query, each answer on a separate line.

Example

xorlove.in	xorlove.out
5 5 1 2 3 4 5 1 2 1 3 2 4 2 5 2 4 5 1 1 7 2 4 3 1 3 4 2 4 1	2 3 3

Note

$\oplus$ — is exclusive OR, e.g. $3 \oplus 5 = 6$.

Problem C. Lumberjack

Input file:	`lumberjack.in`
Output file:	`lumberjack.out`
Time limit:	1 second
Memory limit:	64 megabytes

LumberJack — is a one of the most popular games in Telegram. Rule of this game is simple, Jack with chainsaw stays at the bottom of some tree, using two buttons: left and right he can saw tree. After each saw tree goes down and Jack must avoid tree branches, if left button is pressed then Jack traverses to the left side of tree or stays on left side if he's already on left side. The goal is to saw as much as possible.

Nazerke liked this game. And she wanted to create mobile application with similar mechanism, she called application *Monkey Climb*. In this game monkey climb the tree from downside to upside, user can do this using two buttons: left and right. If pressed button type is same as current monkey position then he goes one level upper, otherwise he goes to opposite side of current level. Also, monkey couldn't go through dangerous zones (in map will be represented as 'x' character). The score of game depends on sequence of pressed buttons: initially $M = 1$, if pressed button is same as previous pressed button then M increases twice, otherwise $M = 1$, after each pressing button M is added to total score.

For given information about tree and dangerous zones find correct sequence of buttons, such that monkey will climb whole tree, also calculate total score for given sequence. For clarification see explanation.

Input

First line contains one integer number n ($1 \le n \le 50$) — number of levels on tree. Next n lines describes tree information from upside to downside, each line contains three characters $left_i$, | and $right_i$ — '.' for safe zone and 'x' for dangerous zone. 'o' for monkey initial position. It's guaranteed that monkey can climb whole tree and initial position is on downside.

Output

On first line output total score and on second line output sequence of pressed buttons.

Example

lumberjack.in	lumberjack.out
6 .|. x|. .|. .|x .|. o|.	17 LLLRRRLL

Note

8

7|6

x|5

3|4

2|x

1|.

o|.

Numbers in cells are position of monkey after each step. Score: $1 + 2 + 4 + 1 + 2 + 4 + 1 + 2 = 17$.

Problem D. The Fate of the Optimization

Input file:	`fate.in`
Output file:	`fate.out`
Time limit:	2.5 seconds
Memory limit:	16 megabytes

Azizkhan, YerzhanU, Mex-mans likes playing some puzzle games. In one of the trip from London to Almaty, Madiyar invented puzzle game called *The Fate of the Optimization* and challenged his geeky friends with that game. Rule of game consists few steps:

1. At first, YerzhanU creates some sequence containing n integers and give it to Mex-mans, lets call that sequence $(a_1, a_2, ..., a_n)$.
2. Then Mex-mans chooses integers l and r, such that $1 \le l \le r \le n$ and gives sequence $(a_l, a_{l+1}, ..., a_r)$ to Azizkhan, lets call this sequence as $(b_1, b_2, ..., b_k)$, where $k = r - l + 1$.
3. Finally, Azizkhan must create permutation $(p_1, p_2, ..., p_k)$ and score of his permutation is $\sum_{i=1}^{k} p_i \times b_i$. Azizkhan must minimize *score* and write it to the paper.

2nd and 3rd step of game is repeated q times. Azizkhan didn't solve logical problems for a while, therefore he didn't handled his part of puzzle. So you're given created sequence of YerzhanU and all q queries of Mex-mans. Help Azizkhan to answer to all queries.

Input

First line contains two integer numbers n and q ($1 \le n \le 10^6$, $1 \le q \le 10^4$) — length of initial sequence and number of queries to Azizkhan. Next line contains n integer numbers a_i ($0 \le a_i \le 10^4$) — i-th number of sequence. Next q lines contains two integer numbers l_i and r_i ($1 \le l_i \le r_i \le n$) — i-th query of Mex-mans to Azizkhan.

Output

For each query print $score_i$ — minimal value which can be calculated by Azizkhan on i-th step. Answers must be printed in separate lines.

Example

fate.in	fate.out
5 3 3 2 1 4 2 1 3 2 5 1 5	10 18 29

Note

Sequence of integer numbers $(p_1, p_2, ..., p_k)$ is a *Permutation*, iff all integers in sequence are different and $1 \le p_i \le k$ for all $1 \le i \le k$.

For the second query, Azizkhan takes from Mex-mans sequence $(2, 1, 4, 2)$ and best permutation is $(3, 4, 1, 2)$ and $score_2 = 2 * 3 + 1 * 4 + 4 * 1 + 2 * 2 = 6 + 4 + 4 + 4 = 18$.

Problem E. Tricolor

Input file:	rgb.in
Output file:	rgb.out
Time limit:	1 second
Memory limit:	64 megabytes

Each Thursday in **Codebusters** is a Football day (or soccer). **Codebusters** has n employees, each of them likes one of the color from: red, green and blue. Each employee came to the football match in their favorite colored t-shirt. And now they need to divide into two team. Division into teams must satisfy conditions below:

- Both teams must have same number of players
- If there is two player with same t-shirt color then they must be in one team

You're given favorite color of each employee, find out is it possible to **Codebusters** employees divide into two teams and play football or they will take another beer today.

Input

First line contains one even integer number n ($2 \leq n \leq 30$) — number of employees in **Codebusters**. Next line contains string s containing exactly n characters of 'R', 'G', 'B' — i-th characters describes favorite color of i-th employee.

Output

On single line print "YES" (without quotes) if it is possible to divide into teams, otherwise print "NO" (without quotes).

Examples

rgb.in	rgb.out
8 BGGRRRBR	YES
6 GRGRGG	NO

Problem F. Chess tournament

Input file:	`chess.in`
Output file:	`chess.out`
Time limit:	1 second
Memory limit:	256 megabytes

Chess is an ancient game with big history. On professional level there is different variation of chess, main differences of them are time that given to each player. There is some fast game variation, like Blitz and long like grandmasters game. All of them has own beauty. Also there is different type of tournaments in chess. Tournaments are seems fairy if each person plays with each, like round-robin. But also exist different type of olympic system, swiss etc, which are not always fairy(IMHO). Main goal of those systems is to determine correct standings after play as less as possible and different systems solves it differently and of course has disadvantages.

Nowadays we want to create new chess tournament format *Kazakh-System*. In this format will held n participants (n is even) and there'll be held exactly k tours with conditions:

- In each tour each participant must play exactly one game with other participant, i.e. there will be $n/2$ games each tour
- No two participant will play more than once in whole tournament
- If person A played with B in tour i, C played with D in tour i and A played with C in tour j then B must play with D in tour j. For all valid persons A, B, C, D

Determine is there exist such arrangements, if so print out schedule of tournament, participants numbered from 1 up to n. If there is multiple solution print out any of them.

Input

Single line of input contains two integers n and k ($1 \le k < n \le 300$) — number of participants and number of tours in tournament.

Output

Print in single line "NO" (without quotes) if there is no solution. Otherwise, on first line print "YES" (without quotes), then print out each tour schedule. Each tour must start with line "Tour x" (without quotes), where x is a tour number, next $n/2$ lines must contain pairs x_i and y_i — players with numbers x_i and y_i will play in i-th game in tour x.

Example

chess.in	chess.out
4 2	YES Tour 1 1 2 3 4 Tour 2 1 4 2 3

Problem G. Codebusters problem

Input file:	codebusters.in
Output file:	codebusters.out
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Aibek experienced backend developer from **Codebusters**. Also, he is well-known as overconfident person. CEO of **Codebusters** Murat brought another client with project for some big money. Task description was very simple, while solution requires to write complex data structure algorithm. In task you need to analyze string S containing lowercase and uppercase letters of english alphabet. For each $1 \le k \le |S|$ you need to find substring of S with maximum length and which is *k-good*. String s called *k-good* if there is exist some another string t and s is equal to concatenation of k copy of t. Aibek has big troubles in algorithm problems. So he asked Temirulan (junior backend developer of **Codebusters**) to help him. Temirulan also didn't solved this problem, but he is one of the problemsetter of *KBTU Open 2017 Spring* and he decided to give problem as one of the problem in contest. Help **Codebusters** backenders to solve this problem.

Input

Input contains only one string S ($1 \le |S| \le 2*10^5$) — input string containing only lowercase and uppercase letters of english alphabet.

Output

For each $1 \le k \le |S|$ print in separate line l and r, such that $r - l + 1$ is maximal and substring $[l, r]$ of S is *k-good*, if there is multiple solution print out with minimal value l, if there is no solution print "-1 -1" (without quotes). Comparison of strings are case sensitive.

Examples

codebusters.in	codebusters.out
aaaaa	1 5 1 4 1 3 1 4 1 5
aacbbb	1 6 1 2 4 6 -1 -1 -1 -1 -1 -1

Note

Substring of string is concatenation of some consecutive characters of this string. For example, "abac", "cab", "b" are substrings of "abacaba", but "ababa', "abc" are not.

String "abacabac" is *2-good* because it is concatenation of two strings "abac".

Problem H. Collect production

Input file:	collect.in
Output file:	collect.out
Time limit:	2 seconds
Memory limit:	64 megabytes

You're given n integer intervals $[l_i, r_i]$ and integer s. Find sequence $(a_1, a_2, ..., a_n)$, such that:

- $l_i \le a_i \le r_i$ for all $1 \le i \le n$
- $a_1 \times a_2 \times ... \times a_n = s$, i.e. product of all numbers in sequence must be s

Print out any valid sequence or identify that there is no such sequences.

Input

First line contains two integer numbers n and s ($1 \le n \le 10^5$, $1 \le s \le 10^9$). Each of the next n lines contains two integer numbers l_i and r_i ($1 \le l_i \le r_i \le 10^9$) — i-th interval.

Output

First line print "YES" (without quotes) if there exist solution, then on next line print out valid sequence $(a_1, a_2, ..., a_n)$ separated by single space, if there is multiple solution you can print any of them. If there is no solution print in single line "NO" (without quotes).

Examples

collect.in	collect.out
4 24 3 10 2 5 1 3 2 6	YES 6 2 1 2
4 24 3 10 6 6 1 2 4 6	NO

Problem I. Another strange sequence

Input file:	`sequence.in`
Output file:	`sequence.out`
Time limit:	1 second
Memory limit:	512 megabytes

Damir invented another interesting sequence. Let's define two functions:

- $f(x)$ — number of different sequences $(b_1, b_2, ..., b_k)$, such that $b_i \leq b_{i+1}$ for all $1 \leq i < k$, $b_1 + b_2 + ... + b_k = x$ and each value in sequence must be power of 2. Two sequence is different iff sizes are different or exist index where two sequences has different values.
- $g(x) = f(1) + f(2) + ... + f(x)$.

You're given T queries, each contains one integer n. For each query output value of $g(2^n - 1)$ modulo 1000000007 $(10^9 + 7)$.

Input

First line contains one integer number T $(1 \leq T \leq 2000)$ — number of queries. Each of the next T lines contains one integer number n $(1 \leq n \leq 2000)$.

Output

Output T lines, in i-th line print out one integer number — answer to the i-th query.

Example

sequence.in	sequence.out
3 1 2 5	1 5 1625

Note

$f(1) = 1$ — there is only one sequence (1).

$f(2) = 2$ — there is two sequences (1; 1), (2).

$f(3) = 2$ — there is two sequences (1; 1; 1), (1; 2).

$g(2^2 - 1) = g(3) = f(1) + f(2) + f(3) = 1 + 2 + 2 = 5$.

Problem J. Arlan jumping

Input file:	`jumper.in`
Output file:	`jumper.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

KBTU Arlans — are one of the famous ACM team from KBTU. Members of team are Aidos, Pernekhan and Temirulan. They prepared a lot, participated in many training contests and camps. As they say, best training camp is Petrozavodsk Camp.

Training a lot not so funny as participating in real contest, training requires more energy. So usually in camps there are holds some activities, where participants can relax from regular hard contest. *Arlans* was inspired by this idea of relaxing and uses it as well by own, not only on camps. Once, between trainings they planned to visit *Tornado* archipelago, i.e. group of islands on sea. This archipelago contains n islands and there is m bridges between some of them. Lets enumerate islands from 1 up to n. We can traverse through bridges in both direction, i.e. graph is undirected. Even on a rest *Arlans* still solves some interesting problems, if there is no interesting problems they invent it by themselves. When they arrived archipelago Temirulan was surprised by the beauty and offered to visit **each** island, then Pernekhan said that they can traverse from island to island via bridge and take photo each time, but for Aidos it was too boring to traverse one by one. So he offered some complicated plan:

1. Let define starting island ID as v and s is equal to 1
2. They took photo on island v
3. Then they choose any preferred valid island u, such that from island u is reachable from island v in s steps
4. Moves to island u, i.e. $v = u$ and double value of s, i.e. $s = 2 * s$.
5. Repeat from step 2

Island u is reachable from island v in k steps if there is exist sequence of numbers $(p_1, p_2, ..., p_{k+1})$, such that $1 \le p_i \le n$ for all $(1 \le i \le k + 1)$, $p_1 = v$, $p_{k+1} = u$ and for each $1 \le i \le k$ there're exist bridge between island p_i and island p_{i+1}.

Aidos: We can traverse through bridges for infinite time (*Tornado* archipelago is really cool) and we need to took photo on **each** island.

Temirulan: Ok, lets choose starting vertex and build our moving plan

Pernekhan: Stop, is it possible to took photo on all island?

Aidos: Oh, yes. It's not guaranteed that we can.

Temirulan: Maybe we'll construct some additional bridges?

Pernekhan: Why not? Lets do this.

Aidos: Ok, but I don't want to work too much. So lets construct minimum number of bridges that are required to reach our goal.

Help *Arlans* to took photo on all islands and find out minimum number of bridges that are required and list of all bridges.

Input

First line contains two integer n and m ($1 \le n \le 10^6$, $0 \le m \le min(10^6, n * (n - 1)/2)$) — number of islands and number of existed bridges. Next m lines contains two integer numbers v_i and u_i ($1 \le v_i, u_i \le n$, $v_i \ne u_i$) — i-th bridge connecting islands with number v_i and u_i. It's guaranteed that there is no more than one bridge between any pair of islands.

Output

At first line print one integer number k — minimum number of constructing bridge. In next k lines print two integer numbers v_i and u_i ($1 \le v_i, u_i \le n$, $v_i \ne u_i$) — new bridge from u_i to v_i. After adding new bridges there must be still no more than one bridge between any pair of islands.

Example

jumper.in	jumper.out
4 4 1 2 2 3 3 4 4 1	1 1 3

Note

There is no possibilities to reach goal without constructing additional bridges. They start from vertex 1, $s = 1$. They took photo on island 1. They move to island 2, $s = 2$. They took photo on island 2. They move in 2 steps to island 4 (path is 2->3->4), $s = 4$. Took photo on island 4. They move in 4 steps to island 1 (path is 4->3->1->2->1), $s = 8$. They already has photo on island 1. Finally they move in 8 steps to island 3 (path is 1->2->3->1->4->1->2->1->3), $s = 16$. They'll took photo on island 3 and they reached their goal.

Problem K. Avoiding permutation

Input file:	`avoid.in`
Output file:	`avoid.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Given sequence with n integers $(a_1, a_2, ..., a_n)$ and permutation of size 3 (p_1, p_2, p_3). Find triple (x, y, z), such that:

- $1 \le x < y < z \le n$
- $(a_x - a_y) * (p_1 - p_2) > 0$
- $(a_y - a_z) * (p_2 - p_3) > 0$
- $(a_z - a_x) * (p_3 - p_1) > 0$
- And among all triples that are satisfied all above conditions choose triple with minimum value of $a_x + a_y + a_z$

Print out $a_x + a_y + a_z$ or tell that there is no such triples.

Input

First line contains one integer number n $(1 \le n \le 10^6)$ — size of sequence. Next line contains n integer numbers a_i $(1 \le a_i \le 10^6)$ — i-th number of sequence. Last line contains three integer numbers p_1, p_2 and p_3 — permutation of size 3.

Output

Print one integer number — answer to the problem, if there is no such triples output "-1" (without quotes).

Example

avoid.in	avoid.out
5 2 3 1 4 4 2 1 3	7

Note

There are two best answers for (x, y, z): $(1, 3, 4)$ and $(1, 3, 5)$, you can check that by your own :)

Problem L. Everything important

Input file:	`april-fool-day.in`
Output file:	`april-fool-day.out`
Time limit:	1 second
Memory limit:	64 megabytes

Everybody knows about April Fool Day. This is a day when everyone in the world joke. Even big companies like Google, Facebook also jokes. They can present fake products. Few years ago Google presented service which can send smell through configuring cooler settings. Of course it was a joke. There are many interesting jokes. Also there is some thing that are not joke. Like Twitter still has limitation to number of words in one tweet. What you think about that problem, is that ok?

Input

One line contains one integer number n ($1 \le n \le 10$).

Output

Output one integer number — answer to problem.

Examples

april-fool-day.in	april-fool-day.out
4	5
5	14