Problem A. Hyperhuffman

Input file: `huffman.in`
Output file: `huffman.out`

You might have heard about Huffman encoding — that is the coding system that minimizes the expected length of the text if the codes for characters are required to consist of an integral number of bits.

Let us recall codes assignment process in Huffman encoding. First the *Huffman tree* is constructed. Let the alphabet consist of N characters, i-th of which occurs P_i times in the input text. Initially all characters are considered to be active nodes of the future tree, i-th being marked with P_i. On each step take two active nodes with smallest marks, create the new node, mark it with the sum of the considered nodes and make them the children of the new node. Then remove the two nodes that now have parent from the set of active nodes and make the new node active. This process is repeated until only one active node exists, it is made the root of the tree.

Note that the characters of the alphabet are represented by the leaves of the tree. For each leaf node the length of its code in the Huffman encoding is the length of the path from the root to the node. The code itself can be constrcuted the following way: for each internal node consider two edges from it to its children. Assign 0 to one of them and 1 to another. The code of the character is then the sequence of 0s and 1s passed on the way from the root to the leaf node representing this character.

In this problem you are asked to detect the length of the text after it being encoded with Huffman method. Since the length of the code for the character depends only on the number of occurences of this character, the text itself is not given — only the number of occurences of each character. Characters are given from most rare to most frequent.

Note that the alphabet used for the text is quite huge — it may contain up to 500 000 characters.

Input

The first line of the input file contains N — the number of different characters used in the text ($2 \le N \le 500\,000$). The second line contains N integer numbers P_i — the number of occurences of each character ($1 \le P_i \le 10^9$, $P_i \le P_{i+1}$ for all valid i).

Output

Output the length of the text after encoding it using Huffman method, in bits.

Example

huffman.in	huffman.out
3 1 1 4	8

Problem B. Robbers

Input file: `robbers.in`
Output file: `robbers.out`

N robbers have robbed the bank. As the result of their crime they chanced to get M golden coins. Before the robbery the band has made an agreement that after the robbery i-th gangster would get X_i/Y of all money gained. However, it turned out that M may be not divisible by Y.

The problem which now should be solved by robbers is what to do with the coins. They would like to share them fairly. Let us suppose that i-th robber would get K_i coins. In this case unfairness of this fact is $|X_i/Y - K_i/M|$. The total unfairness is the sum of all particular unfairnesses. Your task as the leader of the gang is to spread money among robbers in such a way that the total unfairness is minimized.

Input

The first line of the input file contains numbers N, M and Y ($1 \le N \le 1000, 1 \le M, Y \le 10000$). N integer numbers follow - X_i ($1 \le X_i \le 10000$, sum of all X_i is Y).

Output

Output N integer numbers — K_i (sum of all K_i must be M), so that the total unfairness is minimal.

Example

robbers.in	robbers.out
3 10 4 1 1 2	2 3 5

Problem C. Beloved Sons

Input file: `beloved.in`
Output file: `beloved.out`

Once upon a time there lived a king and he had N sons. And the king wanted to marry his beloved sons on the girls that they did love. So one day the king asked his sons to come to his room and tell him whom do they love.

But the sons of the king were all young men so they could not tell exactly whom they did love. Instead of that they just told him the names of the girls that seemed beautiful to them, but since they were all different, their choices of beautiful girls also did not match exactly.

The king was wise. He did write down the information that the children have provided him with and called you, his main wizard.

"I want all my kids to be happy, you know," he told you, "but since it might be impossible, I want at least some of them to marry the girl they like. So please, prepare the marriage list."

Suddenly you recalled that not so long ago the king told you about each of his sons, so you knew how much he loves him. So you decided to please the king and make such a marriage list that the king would be most happy. You know that the happiness of the king will be proportional to the square root of the sum of the squares of his love to the sons that would marry the girls they like.

So, go on, make a list to maximize the king's happiness.

Input

The first line of the input file contains N — the number of king's sons ($1 \le N \le 400$). The second line contains N integer numbers A_i ranging from 1 to 1000 — the measures of king's love to each of his sons.

Next N lines contain lists of king's sons' preferences — first K_i — the number of the girls the i-th son of the king likes, and then K_i integer numbers — the girls he likes (all potentially beautiful girls in the kingdom were numbered from 1 to N, you know, beautiful girls were rare in those days).

Output

Output N numbers — for each son output the number of the beautiful girl he must marry or 0 if he must not marry the girl he likes.

Denote the set of sons that marry a girl they like by L, then you must maximize the value of

$$\sqrt{\sum_{i \in L} A_i^2}$$

Example

beloved.in	beloved.out
4 1 3 2 4 4 1 2 3 4 2 1 4 2 1 4 2 1 4	2 1 0 4

Problem D. Nonoptimal Assignments

Input file: `assign.in`
Output file: `assign.out`

You might have heard about *assignment problem*, stated as follows. Given $n \times n$ matrix of integer numbers, one has to choose n elements of the matrix, so that for each row and each column exactly one element is selected from it, and the sum of the selected elements is minimal possible.

Little Mini has just heard about the problem and thinks that it can be solved using so called "greedy algorithm". That is, she thinks that it is possible to take the smallest element from the first row, after that take minimal element from the second row, that does not belong to the column that was already used, and so on. Each time if there are several possible elements, the one from the smallest column is used.

Her brother Maxi understands that it is not true. To prove it, he wants to construct a matrix where Mini's assignment would be non-optimal. Help him to do so.

Input

Input file contains n ($2 \le n \le 100$).

Output

Output $n \times n$ matrix of integer numbers where Mini's algorithm would not find the optimal assignment. If no such matrix exists, output "`Impossible`" instead. Matrix must contain only non-negative integer numbers not exceeding 100.

Example

assign.in	assign.out
2	0 1 1 10

Problem E. Oil Deal

Input file: `oil.in`
Output file: `oil.out`

Oil is a very important strategic resource. Recently United States of Antarctica invaded the rich in oil country of Qari, and now try to keep the control of the oil transportation system. The system consists of pipelines that connect different nodes — oil sources and main country cities and ports. It is designed in such a way that it is possible to transport oil from any node to any other one.

However the resisting native forces of Qari are not satisfied with the situation. So they continuously perform terror acts, blowing up some oil pipelines. Recently terrorists have decided to perform a series of explosions and want to hurt the oil pipeline system as much as possible.

For each pipeline the terrorists know the cost of blowing it up. They have a fixed sum of money and want to explode as many pipes as possible for this sum. However, since they still need oil for themselves in different regions of the country, they want the system still be able to transport oil from any node to any other one. Help them to find out which pipes to blow up.

Input

The first line of the input file contains n — the number of nodes, m — the number of pipelines, and s — the amount of money terrorists have ($2 \le n \le 50\,000$, $1 \le m \le 100\,000$, $0 \le s \le 10^{18}$). The following m lines contain information about pipelines — for each pipeline the nodes it connects and the cost of blowing it up is specified (cost does not exceed 10^9).

Oil can be transported along each pipeline in both directions, each two nodes are connected by at most one pipeline.

Output

On the first line of the output file print the maximal number of pipelines terrorists can blow up. On the second line print the numbers of these pipelines (pipelines are numbered starting with 1 as they are listed in the input file).

Example

oil.in	oil.out
6 7 10 1 2 3 1 3 3 2 3 3 3 4 1 4 5 5 5 6 4 4 6 5	2 1 5

Problem F. Beautiful Permutation

Input file: `perm.in`
Output file: `perm.out`

Consider a permutation of integer numbers from 1 to n. Let us call length of the longest monotonic subsequence of the permutation its *ugliness*.

For example, the ugliness of the permutation $\langle 1, 2, 5, 3, 4 \rangle$ is 4 because it has a monotonic subsequence $(1, 2, 3, 4)$ of length 4, but has none of length 5. The ugliness of the permutation $\langle 5, 6, 3, 4, 1, 2 \rangle$ is 3 because it has a monotonic subsequence $(5, 3, 1)$ of length 3.

Let us call *beautiful* those permutations that have a smallest possible ugliness for given n. Given n, you must find the first in lexicographic order beautiful permutation of size n.

Input

Input file contains n ($1 \le n \le 10\,000$).

Output

Output the first in lexicographic order beautiful permutation of size n.

Example

perm.in	perm.out
2	1 2
4	2 1 4 3

Problem G. Bandits

Input file: bandits.in
Output file: bandits.out

After robbing a caravan on a road, m bandits have obtained a loot of n similar diamonds. They decided to divide the loot. To do so they have ordered themselves by their birthdate, and make proposals in turn.

When the proposal is made, the bandits vote either for, or against it. If the strict majority of bandits votes for the proposal, it is accepted, in the other case the author of the proposal is killed.

The proposal states for each of the still alive bandits, what number of diamonds should he get. All bandits are smart, greedy, bloodthirsty and careful. That means, that the bandit votes against the proposal if and only if he is certain that he will get the same number of diamonds or more in future assuming that all the other bandits also vote optimally.

Find out what is the maximal number of diamonds the youngest bandit (the one who makes the first proposal) can get. If he is killed regardless of his proposal, output “-1”.

Input

Input file contains m and n ($1 \le m \le 2000$, $1 \le n \le 2000$).

Output

Print the maximal number of diamonds the first bandit can get, or “-1” if he is killed regardless of his proposal.

Example

bandits.in	bandits.out
5 1000	997
2 1000	-1

Problem H. Princess Dilemma

Input file: princess.in
Output file: princess.out

Once upon a time in the kingdom far away there lived a princess. And she was so beautiful that every young man in the kingdom wanted to marry her. But the princess was very haughty and didn’t want to marry the first man she meets. Therefore she decided that she would marry only the most beautiful young man in the kingdom.

But there was a little problem — how to select the most beautiful one? Of course, had all young people gathered together in front of the princess, she would immediately tell who is the most beautiful. But the young men are usually rather busy playing, fighting, and doing other very important things. Therefore all she could was to ask them to come one person a day, and choose the one to marry.

Fortunately, the princess had a very good memory. After seeing a man she could always tell for any other man, whether he was more beautiful, or less beautiful. The princess beauty criteria were so exact, that no two young men in the kingdom were equally beautiful for her.

The princess decided to do the following. She created the list of all n beautiful young men and ordered it randomly. After that she would invite young men to her castle one after another in the order of the list. After meeting each man she would decide whether she would marry him, or not. After she married one man, she would continue to invite other young men to make sure she married the most beautiful one. If it would indeed be so, they would live happily until the end of the days. In the other case, if she married the man who was not the most beautiful, or didn’t marry at all, the princess would commit suicide.

Help the princess to develop the best strategy, so that she marries the most beautiful young man with the maximal possible probability. The only thing that the princess can do to make a decision whether to marry the current young man is to compare his beauty to the beauty of all previous young men.

Input

The input file contains one integer number n ($1 \le n \le 2006$).

Output

Output one real number — the probability that the princess would marry the most beautiful young man if she follows the optimal strategy. The probability is taken over all possible young men orderings. The princess strategy must be deterministic.

Your answer must be accurate up to 10^{-8}.

Example

princess.in	princess.out
1	1.0
2	0.5
3	0.5

In the last example the princess must act as follows. The first young man must be rejected. The second young man must be accepted if and only if he is more beautiful than the first one. Using this strategy the princess fails to marry the most beautiful young man for the following permutations of beauty (the greater — the more beautiful): $\langle 1, 2, 3\rangle$, $\langle 3, 1, 2\rangle$, $\langle 3, 2, 1\rangle$, and succeeds for all other permutations. All other strategies give a smaller success probability.

Problem I. Lempel-Ziv Compression

Input file: lz.in
Output file: lz.out

Most modern archivers, such as WinRAR or WinZIP, use modifications of Lempel-Ziv method as their primary compression algorithm. Although decompression of LZ-compressed archives is usually easy and fast, the compression process itself is often rather complicated and slow. Therefore professional archivers use approximation methods that sometimes do not allow to achieve the best possible compression.

This situation doesn’t satisfy your chief George. He would like to create the best archiver WinGOR. The archiver will use the following modification of LZ77 algorithm.

The text is partitioned to chunks of length not exceeding 4096. Each chunk is compressed independently. We will describe the decompression of one chunk t. Based on this description, you will have to create a compression algorithm that will create the shortest possible compressed chunk x from the given chunk t.

The compressed chunk is written down as the sequence of *plain characters* and *repetition blocks*. Plain character is 8 bits long. When decompressing, plain character c is simply copied to output. Repetition block (r, l) consists of two parts: *reference* r and *length*

l, each 12 bits long. Reference r is an integer number between 1 and 4095. When repetition block (r, l) is obtained after decompressing $i-1$ characters of text, characters $t[i-r \dots i-r+l-1]$ are copied to output. Note, that r can be less than l, in this case recently copied characters are copied to output as well.

To help decompressor distinguish between plain characters and repetition blocks a leading bit is prepended to each element of the compressed text: 0 means plain character follows, 1 — repetition block follows.

For example, "`aaabbaaabababababab`" can be compressed as "`aaabb(5,4)(2,10)`". The compressed variant has $8 + 8 + 8 + 8 + 8 + 24 + 24 + 7 = 95$ bits instead of 152 in the original text (additional 7 bits are used to distinguish between plain characters and repetition blocks).

Given a text chunk, find its compressed representation which needs fewest number of bits to encode.

Input

Input file contains a text chunk t. Its length doesn't exceed 4096. A text chunk contains only small letters of the English alphabet.

Output

Print the length of the compressed text in bits at the first line of the output file. Print the compressed chunk itself at the second line of the output file. Use characters themselves to denote plain characters and "`(r,l)`" notation (without spaces) to denote repetition blocks.

Example

lz.in	lz.out
aaabbaaabababababab	95 aaabb(5,4)(2,10)

Problem J. New Year Tree Transportation

Input file: `tree.in`
Output file: `tree.out`

People of Byteland celebrate New Year. Unlike people of most other countries, they do not decorate fir-trees for the New Year celebration. Instead they decorate binary trees. A binary tree is a rooted tree such that every node has at most two children.

A nice binary tree with n nodes was prepared to be set up on the main square of Byteland capital. However first it must be transported from the place where it was grown up to the capital. The transportation will be arranged by the railroad. But it turned out that the standard railroad car can carry the tree only if it has at most k nodes.

So it was decided to cut several edges of the tree so that each of the remaining connected parts had at most k nodes. After the tree is transported to the capital it would be reassembled and set up. Due to security reasons each car must carry only one tree part.

Of course the department of transportation of Byteland would like to use as few cars as possible to transport the tree. However, minimizing the number of cars seemed to be too difficult problem. Therefore the minister of transportation ordered to cut the tree in such a way that the number of cars needed at least did not exceed $\lceil 2n/k \rceil$.

But the people who are transporting the tree couldn't solve even this problem. Help them! Given a binary tree find the way to cut some of its edges in such a way that each of the remaining connected parts had at most k nodes and the number of parts didn't exceed $\lceil 2n/k \rceil$.

Input

The first line of the input file contains n — the number of nodes of the tree, and k — the maximal capacity of the car ($1 \le n \le 100\,000$, $1 \le k \le n$). The following $n-1$ lines describe the edges of the tree. Each edge is described by two integer numbers: the parent node and the child node. The given tree is guaranteed to be a binary tree. Nodes are numbered from 1 to n, root has number 1.

Output

The first line of the output file must contain l — the number of edges that must be cut. The second line must contain l integer numbers — the edges to be cut. Edges are numbered from 1 to $n-1$ as they are listed in the input file.

If the tree cannot be cut in the described way, output $l = -1$.

Example

tree.in	tree.out
5 2 1 2 1 5 5 3 5 4	2 2 4

Problem K. Agrarian Reform

Input file: `agrarian.in`
Output file: `agrarian.out`

The King of Squaredom is planning the agrarian reform. The Squaredom has the form of rectangle of $m \times n$ squares. Squares are identified by pairs (x, y) where x ranges from 1 to m, and y ranges from 1 to n. Each square is either occupied by a peasant's house, or contains a swamp, or is a field. The King would like to assign peasants to fields, so that each peasant was assigned to exactly one field, and each field was assigned as most one peasant.

The King asked his Minister of Agronomy to prepare the list of peasants. After that he would assign them to fields. The Private Counselor of the King has found out the algorithm the King will use to assign peasants to fields.

The King would look through the peasants in order they are listed by the Minister of Agronomy. For each peasant he would find the closest to his house field that has no peasant assigned to it yet. That field would be assigned to this peasant. If there are several such fields, the field which has the smallest x will be chosen, if there are still several such fields, the field which has the smallest y among them will be chosen. The distance between squares (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

The Minister of Agronomy would like to order peasants in such a way that the sum of distances between peasant and the field he is assigned to for all peasants were as small as possible. Help him to find such order.

Input

The first line of the input file contains four integer numbers: m, n, k and s — the size of the field, the number of peasants, and the number of swamps, respectively ($1 \le m, n \le 20$, $1 \le k \le mn/2$, $0 \le s \le mn - 2k$). The following k lines contain coordinates of squares where peasants live, the i-th of these lines contains two integer numbers x_i, y_i ($1 \le x_i \le m$, $1 \le y_i \le n$). No two peasants live in the same square.

The following s lines contain coordinates of squares containing swamps.

Output

Output k numbers — the order in which the Minister of Agronomy should order peasants so that the King assigned them to the fields in the optimal way.

Example

agrarian.in	agrarian.out
3 5 5 0 2 3 2 4 1 3 2 2 3 3	3 4 2 1 5

Problem L. War Academy

Input file: `academy.in`
Output file: `academy.out`

The War Academy of Flatland has received the request from the government to prepare n special agents for the secret operation against its neighbor Edgeland. Each agent will have a special mission assigned, the i-th mission requires the *experience* of an agent that will perform this mission to be at least p_i.

The main instructor of the academy has selected n students for the program. The i-th of the selected students will be assigned to the i-th mission. Now the instructor needs to plan the training sessions.

Each student can have individual training, and all students together can have group training. Initially all students have experience equal to 0. The i-th student is characterized by his *intelligence* q_i and his *sociality* s_i. After taking individual training for m hours the experience of the i-th student increases by mq_i. After taking group training for m hours the experience of the i-th student increases by ms_i. Training can continue for any non-negative number of hours (and can be non-integer if needed).

Training is performed by the instructors. The instructor must be paid a Flatland dollars per hour of individual training and b Flatland dollars per hour of group training.

Help the main instructor of the academy to plan the training in order to fulfil the request of the government and spend as little money as possible.

Input

The first line of the input file contains three integer numbers: n ($1 \le n \le 100$), a and b ($1 \le a, b \le 10^6$). The following n lines contain three integer numbers each: p_i, q_i and s_i ($1 \le p_i, q_i, s_i \le 1000$).

Output

The first line of the output file must contain one real number: w — the total cost of preparing students for the operation, in Flatland dollars. The second line must contain one real number g — the length of group training in hours. The third line must contain n real numbers — for each student specify the length of his individual training in hours.

Output as many digits after the decimal point as possible (the more you output, the less is the chance that the verifying program will have precision errors). The verifying program will use threshold of 10^{-6} when making comparisons of real numbers.

Example

academy.in	academy.out
2 50 13 10 10 2 3 5 1	59.0 3.0 0.4 0.0