Problem A. Protoss Defense

Input file:	`defense.in`
Output file:	`defense.out`
Time limit:	1 second
Memory limit:	256Mb

General High Templar wants to relieve The World of disgusting invaders. His right hand, military commander Tryt o'Kill is ready to perform the plan of the destruction of the enemies. Plan is as simple as brilliant... To locate and to annihilate all enemy military targets!

So, the military commander Tryt o'Kill has been ordered to destroy enemy creatures. He has decided to provide missile attack on the enemy targets. But this task is not so simple... These targets are heavily defended by K different layers of defense. The first layer consists of *forward air defense sites* (FADS), the second layer consists of *band surface to air missiles* (BSAM), the third layer consists of *airborne interceptors* (AI), the other layers consists of different kinds of *terminal surface to air missiles* (TSAM); a $(K+1)$-th layer contains the military targets themselves. Of course, some targets might not be defended while some others might have all defense layers.

Tryt o'Kill is good at math and he has denoted by S the set of all defense sites and by T the set of the enemy targets. A defense site might also provide protection to other defense sites in higher-numbered layers. Some defense sites might protect neither targets nor other defense sites. Commander also denoted by D_i the set of defense sites that protect the target or defense site $i \in T \cup S$. Let L_i denotes the layer's number of $i \in T \cup S$. Of course,

$$\forall\, i \in T \cup S \quad \forall\, j \in D_i \quad \Rightarrow \quad L_i > L_j.$$

Based on his past experience, Tryt o'Kill feels that while it might be possible for missiles to pass through all the defenses and reach the military targets, the probability of such a "leakage" is quite small. Instead, he believes that to destroy a target, he must first destroy all the defense sites that protect it. Therefore, he must destroy defense sites as well as targets. Destroying the i-th target or defense site has a certain military benefit and some loss. So, Commander knows the so called "cost-effectiveness" c_i of destroying the i-th target or defense site. The military commander wants to identify a set of targets and defense sites with the largest possible total "cost-effectiveness." It might be possible that providing missile attack is unreasonable and it would be better to choose nothing.

Input

Let's suppose that all targets and defense sites are numbered from 1 to n, where $n = |S \cup T|$. In the first line of the input file integer numbers n, m and K are written. The second line contains c_i $(i = 1, 2, \ldots, n)$; moreover, $-10^6 \leqslant c_i \leqslant 10^6$. The third line is made up of L_j $(j = 1, 2, \ldots, n)$ divided by space $(1 \leqslant L_j \leqslant K + 1)$. Next m lines consist of pairs of numbers i and j; each pair (i, j) means that $j \in D_i$ and $L_i > L_j$. Look at the example below to get more evident information. Keep in mind that $0 < n \leqslant 200$, $0 \leqslant K < 200$, $0 \leqslant m \leqslant 40\,000$.

Output

Write the largest possible total "cost-effectiveness" of chosen targets or defense sites in the first line of the output file. Write on the second line the number of targets of defense sites to destroy. On the third line write numbers of chosen targets or defense sites in ascending order. If there are several solutions, output any one of them.

Examples

defense.in	defense.out
5 4 2 5 -3 -4 3 2 3 2 2 1 1 2 4 1 3 3 4 3 5	6 4 1 3 4 5
5 4 2 1 2 1 -5 -5 3 2 2 1 1 2 4 1 3 3 4 3 5	0 0

Problem B. Ancient Knowledge

Input file:	`anc.in`
Output file:	`anc.out`
Time limit:	2 seconds (4 for Java)
Memory limit:	256Mb

Ages ago Protoss army was forced to retrograde movement and to yield to pressure of disgusting uncountable Zerg army. Many of the best Protoss warriors laid down their lives. They were holding position until main part of the colony left this strategic place. It was a big battle loss. Protoss called this place The Lost Temple...

Now The Lost Temple is the place covered by ancient ruins. They all put Protoss in remembrances of unexpected Zerg attack. Protoss scientists have discovered ruins and decoded almost all secret information which was written on the walls and then was partly destroyed. But there is only one thing that they can not understand. It is a formula. The formula is rather simple, but may be very important and the scientists are eager to understand its meaning. So, here it is:

$$S_n(k) = \frac{2^n}{\pi} \int_0^{\pi} \cos^n x \cdot \cos((n-2k)x)\, dx$$

Having the goal to understand the meaning of this formula, they decided to compute it for some fixed values of n and k. They started working with zeal, but Zeal comes only to those who use integral values. So, the scientists decided to drop fractional part of the answer; moreover, they will be able to understand the meaning of the formula even if they know only the remainder of dividing the integral part by some power of two.

Input

The input file contains only three integers: n, k and m, where $0 \leqslant n, k \leqslant 50\,000$, $0 \leqslant m \leqslant 30$.

Output

Write only one integer number — the remainder of dividing the integral part of $S_n(k)$ by 2^m;

Examples

anc.in	anc.out
0 1 2	0
0 0 0	0
1 1 10	1

Problem C. Transportation plan

Input file:	`badek.in`
Output file:	`badek.out`
Time limit:	4 seconds
Memory limit:	256Mb

Protoss army is known to be very manoeuvrable and fast. The army uses ancient knowledge to get the right plan of transportation and to kick off the enemies...

But sometimes it is very difficult to create a transportation plan, because ancient knowledge wasn't geared to big battle areas and to big amounts of postures.

To move the forces fast between the main Protoss base and the point of impact Protoss scientists use very simple rules. First, they divide all battle field into several battle locations. Suppose there is a set V of n locations numbered from 1 to n. Also they call these battle locations "nodes." Second, for each pair of nodes (i, j) $(i, j \in V,\ i \neq j)$ they determine whether it is possible to move forces from the location i to the location j using roads, tunnels, transport etc. and how many battle units it is possible to move from i to j simultaneously. Protoss scientists use word "arc" for each pair of nodes (i, j) if $i \neq j$. This amount they call "the capacity of the arc (i, j)" and denote it by c_{ij}. Moreover, $c_{ij} = 0$ means impossibility of moving any battle unit directly from i to j (there is no arc from i to j). Keep in mind that there can be at most one arc from node i to node j. Suppose there are m arcs with positive capacity. Third, they select two special battle locations $s \in V$ and $t \in V$ $(s \neq t)$. s is the main Protoss Base (the start point where all forces are waiting for the "Attack" command) and t is the point of impact (the end point where the enemies are holding their position). The goal is to transfer as large amount of battle units from the start point to the end point as possible. Battle units can move along any arc with the positive capacity. But they shouldn't stop until they get to the end point and it is impossible to move more than c_{ij} battle units from the node $i \in V$ to the node $j \in V$ along the arc (i, j). All this construction with n nodes, m arcs and capacities they are used to calling "Network." Of course, Protoss scientists know this problem very well. They call it "The Maximum Flow Problem."

It is no wonder that Protoss scientists know how to solve this problem using ancient knowledge. Consider the so-called "flow value" f. Initially, $f = 0$. First, they find any "shortest augmenting path" from s to t. The shortest augmenting path is the sequence of nodes s, i_1, ..., i_p, t with the property of $c_{si_1} > 0$, $c_{i_1 i_2} > 0$, ..., $c_{i_p t} > 0$ and no two identical nodes in the sequence. Word "shortest" means that there is no sequence with smaller number of nodes in it with the same property.

Next, $w = \min\{c_{si_1}, c_{i_1 i_2}, \ldots, c_{i_p t}\}$ is determined and added to the flow value f (w is called "the capacity of the path"). It is clear that $w > 0$ and that is why the flow value is increased by some positive amount. The main step of the algorithm is to construct the so-called "Residual Network." Protoss scientists recalculate capacities of the arcs from the augmenting shortest path using the following rule: They subtract value w from c_{si_1}, $c_{i_1 i_2}$, ..., $c_{i_p t}$ and add value w to the "reversed" arcs $c_{i_1 s}$, $c_{i_2 i_1}$, ..., $c_{t i_p}$ (it is named "the augmentation step"). Of course, after this operation some arcs could disappear (if capacity became zero) while some other arcs could appear (if capacity increased from zero to some positive amount). So, the scientists have new network and they start finding the shortest augmenting path again. If there is no such path, they stop and the value of f is the largest possible number of battle units which could be moved from the main base to the point of impact.

The Protoss scientists don't know which network delivers the maximum number A of the augmentation steps. But they would like to imagine one with the fixed number of nodes, some number of arcs and with the following property: $A \geqslant \lfloor n^3/153 \rfloor$ ($\lfloor x \rfloor$ is the integral part of x).

Input

The input file contains only one integer: n, where $6 \leqslant n \leqslant 100$.

Output

First line of the output file should contain four integers: n, m, s and t ($s \neq t$). Moreover, m should satisfy the condition $0 \leqslant m \leqslant n^2$.

Next m lines describe the structure of the battle field (network), each of these lines should contain three integers i, j and c ($i \neq j$) which mean that there is a possibility to move simultaneously c battle units from location i to location j (arc (i, j) with capacity c). There should be at most one arc from node i to node j. Make sure that $1 \leqslant c \leqslant 10^9$ and integer.

Next line should contain two integers: The maximum amount f of battle units which can be moved from s to t and number A of augmentation steps needed to calculate the transportation plan. Flow value f should be less or equal to 10^9. Number of augmentation steps A should not be less than the number $\lfloor n^3/153 \rfloor$.

In next A lines describe the augmentation steps. First number is the capacity of the found path w (this value was described above), second number l is the length of shortest augmenting path (number of nodes decreased by one) and next $l+1$ numbers show the shortest augmenting path from s to t. Look at the example for more information.

It is guaranteed that the solution exists for each $n \geqslant 6$.

Examples

badek.in	badek.out
6	6 5 1 6 1 2 10 2 3 10 3 4 10 4 5 10 5 6 10 10 1 10 5 1 2 3 4 5 6
7	7 4 1 4 1 2 10 2 4 10 1 3 20 3 4 20 30 2 10 2 1 2 4 20 2 1 3 4

Problem D. Imagination

Input file:	`im.in`
Output file:	`im.out`
Time limit:	2 seconds (5 for Java)
Memory limit:	256Mb

Right now the disgusting Zerg creatures are preparing to ruin The Protoss Mineral Base. Now it is your turn. . .

. . . It was three o'clock in the morning and the Protoss colony was as quiet as a church. Military commander Tryt o'Kill was having his heart in his boots. Disgusting Zerg scientists had discovered new technology of communication based on the secret key. Their commanders were able to exchange messages with appalling regularity without fear of being intercepted and decoded. The new coding algorithm was known to be as stupid as Zerg scientists. They had dropped a brick once again. . . Commander Tryt o'Kill understood that calculating the secret key was as easy as ABC. And you were chosen to write a special program with purpose to get the key.

Zerg very like Fibonacci sequences. They have $n+1$ numbers $f_0, f_1, \ldots, f_n$ and they are able to construct a convolution to get the secret key

$$\widehat{F}_n = \sum_{k=0}^{n} f_k f_{n-k}.$$

The numbers f_k are being calculated in the following way: $f_0 = f_1 = 1$ and

$$f_k = f_{k-1} + f_{k-2}, \quad \text{if } k \geqslant 2.$$

If only it was so simple! Filthy Ultralisk decided to use Optical Jammer Source to add some "jamming" (or "noise") x_k to make the calculations more complex. And the formula really became more complex:

$$F_n = \sum_{k=0}^{n} f_k f_{n-k} x_{n-k}.$$

Traditionally, the "noise" is a sequence of some kind of pseudo-random numbers. If x_0 is fixed then the k-th pseudo-random number will be

$$x_k = (A \cdot x_{k-1} + B) \mod m,$$

where A, B and m are fixed numbers.

High Templar summoned the Dark Archon. He caught a Zerg drone and it said that $0 \leqslant n \leqslant 65\,535$, $0 \leqslant A, B, x_0 \leqslant 100$ and $0 < m \leqslant 10$.

Only after he has tortured the drone, he has told him exact parameters. . . It told that they are all integral and. . . kicked the bucket. . .

Input

At the beginning of the first line of the input file there are five integer numbers separated by a single space. These numbers are given in the following order: n, x_0, A, B, m. All constraints are given above.

Output

You have to write the secret key F_n to the output file (see the examples) at the beginning of the first line.

Examples

im.in	im.out
2 1 9 3 7	19
3 1 9 3 7	28
100 4 2 9 3	64431797813040810974729
239 0 17 22 1	0

Problem E. New Zerg Creature

Input file:	`gcs.in`
Output file:	`gcs.out`
Time limit:	1 second (2 for Java)
Memory limit:	256Mb

It was fog-flying monster which attacked General of the Army Tryt o'Kill... Only Psionic Storm and huge battle experience saved General from ruin...

But now Protoss know about new Zerg creature which is more powerful than Ultralisk and it is able to fly like Mutalisk. Of course, genetic code of the new monster is similar to the Ultralisk's one. Protoss scientists would like to find out all disadvantages of the new monster to get ready to fight him.

With this difficult goal scientists started to work. They wrote genetic code of Ultralisk in the first line (genetic code is the sequence of positive integer numbers, length of a genetic code is the quantity of numbers in it) and genetic code of the new monster in the second line. Now they have to remove some numbers from the first code and some numbers from the second code to get both sequences identical. But after removing some numbers the scientists would like both sequences to be as large as possible (they would like to get the so-called "Largest Common Subsequence"). So, it is a problem to choose right numbers to remove. Moreover, each genetic sequence contains any integer number at most ten times, so, each sequence could contain a lot of different numbers! But never say "die —" you were asked to help Protoss scientists.

Input

First line of the input file contains genetic code of Ultralisk. It consists of at most 10^5 positive integer numbers (all numbers are less of equal to 10^6).

Second line contains the genetic code of the new monster — sequence of numbers with the same properties and constraints.

Numbers in the first line can be separated by any number of space characters as well as in the second line.

Output

Output the length of the Largest Common Subsequence of two codes.

Examples

gcs.in	gcs.out
2 5 6 3 1 4 5 3 2 8 1 3 4	4
1 10 100 1000 1000 1 100 10	2

Problem F. Observer

Input file:	`obs.in`
Output file:	`obs.out`
Time limit:	2 seconds
Memory limit:	256Mb

One never knows where a Protoss observer is situated while digging out very useful information about strategic development of the enemies...

But by now Zerg scientists have developed The Temperature Protector which increases the temperature of air around Zerg base and any Protoss observer will be immediately melted if it comes too close to the base.

Of course, Zerg Protector has its faults. It doesn't work stably, therefore, there are some zones over the base where temperature is rather low; moreover, Protoss observer can travel between some of such low-temperature zones using special low-temperature tunnels.

So, there are n low-temperature zones where an observer can stay without afraid of being melted (these zones are numbered from 1 to n). There are m bidirectional low-temperature tunnels between some pairs of zones (i, j). Each such tunnel has its own temperature.

Protoss scientists would like to develop the technique of traveling between any pair of low-temperature zones using the tunnels. Let's consider the temperature of the path to be equal to the temperature of the hottest tunnel on this path. So, the scientists would like to find the lowest temperature path between any pair of low-temperature zones (i, j). You can ignore the temperature of any zone, because it is rather low. Protoss scientists know that the low-temperature tunnels allow to travel from any node i to any node j.

Input

First line of the input file contains two numbers: n and m, where $2 \leqslant n \leqslant 3000$ and $1 \leqslant m \leqslant 50\,000$. Next m lines are made of triples (a, b, t). Each triple means that there is a bidirectional tunnel between zones a and b and it has temperature $0 \leqslant t \leqslant 10^9$.

All numbers in the input file are integral.

Output

Consider a table of numbers with n rows and n columns. In the i-th row and j-th column there is the temperature of the coldest path between zone i and zone j. If $i = j$ then temperature is zero. You have to output sum of all numbers in such a table.

Example

obs.in	obs.out
3 3 1 2 1 1 3 2 2 3 3	10

Hint: The table is $\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}$; so, $2+2+2+2+1+1 = 10$.

Problem G. Protoss Archive

Input file:	`archive.in`
Output file:	`archive.out`
Time limit:	2 seconds
Memory limit:	256Mb

Protoss scientists have very important information to keep in secret. They decided to put the information into the Protoss Archive...

The information is represented by matrix C with m rows and n columns. Each element c_{ij} of the matrix is some integer; moreover, $0 \leqslant c_{ij} \leqslant 100$. Unfortunately, all numbers are rather different and the scientists want elements c_{ij} of the matrix to be as close to each other as possible. They are used to achieving this objective by the so-called "data scaling:" that is, multiplying each row i by a non-negative constant α_i and dividing each column j by a positive constant $\beta_j \geqslant 1$.

The Protoss Archive will work well if numbers c_{ij} (after scaling) satisfy the condition $l \leqslant c_{ij} \leqslant u$ (for some given integers $0 \leqslant l \leqslant u \leqslant 100$).

Given n, m, l, u and matrix $C_{m \times n}$ you have to find such α_i ($i = 1, 2, \ldots, m$) and β_j ($j = 1, 2, \ldots, n$) that yield $l \leqslant \alpha_i c_{ij}/\beta_j \leqslant u$. Moreover, you have to find such α_i and β_j that $0 \leqslant \alpha_i \leqslant 100$ ($i = 1, 2, \ldots, m$) and $1 \leqslant \beta_j \leqslant 100$ ($j = 1, 2, \ldots, n$).

Input

First line of the input file contains four integer numbers m, n, l, u, where $1 \leqslant n, m \leqslant 100$. Next m lines contain n integers each. They describe matrix C.

Output

If the solution exists, write word "`Yes`" (without quotes) in the first line of the output file. In the second line write α_i ($i = 1, 2, \ldots, m$) — m numbers accurate to four decimal places. In the third line write β_j ($j = 1, 2, \ldots, n$) — n numbers accurate to four decimal places.

If there is no solution, write only word "`No`" (without quotes) in the first line of the output file.

Examples

archive.in
```
3 4 4 7
7 3 6 5
6 7 7 6
7 7 5 8
```
archive.out
```
Yes
1.33333333 1.00000000 1.00000000
1.33333333 1.00000000 1.14285714 1.14285714
```
archive.in
```
3 4 4 6
7 3 6 5
6 7 7 6
7 7 5 8
```
archive.out
```
No
```

Problem H. Reserve

Input file:	`reserve.in`
Output file:	`reserve.out`
Time limit:	1 second
Memory limit:	256Mb

Protoss prepare a new attack. They have a lot of battle units to attack, but it is necessary to leave some Zealots (it is one of the most useful Protoss battle unit) on the base (as a reserve) to be sure that the base is protected from any counterattack…

One Zealot requires c_{ij} minerals to defend the base if employed from day i to day j. Attack operation proceeds for n days. Protoss commander Tryt o'Kill calculated that in day k he needs at least b_k Zealots on the base ($k = 1, 2, \ldots, n$).

The general can keep exactly b_k units on the base and pay them required number of minerals, but he wants to minimize the amount of minerals spent during all n days. The general can also keep more than b_k units on day k if this will result in smaller total amount of minerals.

Input

First line of the input file contains number of days n, $1 \leqslant n \leqslant 50$. Next n lines describe cost of keeping Zealots. So, j-th number in the $(i+1)$-th line of the input file is $c_{i,i+j-1}$. All costs are non-negative and don't exceed 10 000. The last line contains b_k ($k = 1, 2, \ldots, n$). Consider $0 \leqslant b_k \leqslant 2000$.

All numbers in the input file are integral.

Output

In the first line write total amount of minerals General has to pay to Zealots to defend the base. Output a triangular table: j-th number of the $(i+1)$-th line should contain number of Zealots which have to defend the base from day i to day $i+j-1$. This number shouldn't exceed 10^9. Keep in mind that Protoss General wants to minimize amount of minerals paid to Zealots.

It is guaranteed that the optimal amount of minerals doesn't exceed 10^9.

Example

reserve.in	reserve.out
3 3 2 3 3 2 3 2 3 2	7 0 1 1 0 1 0

Problem I. Reflection

Input file:	`ref.in`
Output file:	`ref.out`
Time limit:	2 seconds (4 for Java)
Memory limit:	256Mb

Protoss scientists have developed Psyonic-Total-Reflection-Prism and called it “NewPylon.” Now instead of building an old Pylon Protoss Probes always build “NewPylon,” and for simplification they call it Pylon, too. This new Pylon has a very important advantage: it can be teleported by reflection in a plane. In other words, teleporting a Pylon means reflecting it with respect to a special reflection plane.

Any Pylon has $1 \leqslant n \leqslant 10^5$ base points. Actually, teleporting the Pylon is the same as teleporting its base points. Protoss have $0 \leqslant m \leqslant 10^3$ reflection planes. Each plane is given by three points on it (these points are noncollinear). To get the Pylon teleported Protoss have to reflect each base point with respect to each plane consistently: to reflect with respect to the first plane, then the second and so on for each base point.

Given the points on the reflection planes and base points of the Pylon you have to calculate reflected points of the Pylon.

Input

First line of the input file contains integer m. Next m lines contain nine integer numbers each: x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3 and z_3. First three integers are the coordinates of the first point, next three integers are the coordinates of the second point on the plane, and the last three integers are the coordinates of the last point on the reflection plane. As it was said these three points always define a plane because they are noncolinear.

Next line contains number n. The last n lines describe the Pylon. They contain three integers each: x, y and z — basic point of the Pylon.

Coordinates of any point don’t exceed 100 by the absolute value.

Output

Write n lines. Each line should contain coordinates of the reflected base point of the Pylon. The points should go in the same order as they were given in the input file. Output coordinates to an accuracy of four decimal places.

ref.in	ref.out
1 1 0 0 0 1 0 0 0 1 1 0 0 0	0.6667 0.6667 0.6667
1 1 0 0 0 1 0 0 0 1 1 1 0 0	1.0000 0.0000 0.0000
2 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 1 2 0 0 0 1 2 3	0.6667 0.3333 0.3333 -2.3333 1.3333 2.3333

Problem J. kD Cube

Input file: cube.in
Output file: cube.out
Time limit: 2 seconds
Memory limit: 256Mb

It is well known that any integer number n can be presented as a sum $n = a^2 + b^2 + c^2 + d^2$, where a, b, c and d are some integer numbers.

High Templar has one integer number n; moreover $1 \leqslant n \leqslant 100\,000$. He wants to represent this number as a sum of powers of positive integer numbers:

$$n = \sum_{j=1}^{m} a_j^k$$

for $k = 2, 3, \ldots, 10$. But there is a problem: High Templar wants to minimize m — number of summands. No wonder that you have to help him...

Input

The input file contains only integer number n.

Output

Write nine lines into the output file. First number in each line is the minimal number of summands (it is m). The following m numbers are a_j ($j = 1, 2, \ldots, m$). For the first line $k = 2$, for the second one $k = 3$ and so on. For the last line $k = 10$. Look at the example below.

Examples

cube.in	cube.out
17	2 1 4 3 1 2 2 2 1 2 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1024	1 32 2 8 8 4 4 4 4 4 1 4 16 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 8 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 1 2

Hint: In the first example $17 = 1^2 + 4^2$ (two summands), or $17 = 1^3 + 2^3 + 2^3$ (three summands) etc.