Bikes vs Cars

Problem Name	Bikes vs Cars
Time Limit	5 seconds
Memory Limit	1 gigabyte

In Lund is fietsen een heel gewone manier van verplaatsen. Fietsen en auto's samen passen moeilijk in smalle straatjes. Om dit verbeteren wil de gemeente het hele stratennetwerk opnieuw indelen.

Er zijn N belangrijke locaties (genummerd van 0 tot $N-1$) waartussen mensen in Lund regelmatig reizen. Mensen reizen tussen twee locaties door een route te volgen, door verschillende straten. Een voertuig (auto of fiets) past over een route als alle relevante rijstroken breed genoeg zijn. Elke nieuw-gelegde straat verbindt twee van deze belangrijke locaties en heeft een totale breedte van W. Deze breedte kan vrij verdeeld worden tussen een fiets-strook en een auto-strook. In Lund hebben techneuten recent fietsen en autos van breedte 0 uitgevonden (deze kunnen op rijstroken van breedte 0 rijden).

De techneuten hebben de breedtes van de auto's en fietsen in de stad gemeten. Voor elk paar belangrijke locaties, weten ze de breedste auto en de breedste fiets dat tussen deze locaties kan reizen. De gemeente eist ook dat er geen bredere auto's of fietsen tussen deze twee locaties kunnen reizen.

Voor elk paar i, j ($0 \leq i < j \leq N-1$) krijg je twee gehele getallen $C_{i,j}$ and $B_{i,j}$. Jouw taak is het om een stratennetwerk te ontwerpen dat de N locaties verbindt. De straten hebben allemaal een breedte van W, maar voor elke straat s kun je de breedte van de fietsstrook bepalen b_s en dit bepaalt de breedte van de autostrook $W - b_s$. Het network moet voldoen aan de volgende voorwaarden:

- Je moet tussen elk paar locaties kunnen reizen. Het kan zijn dat je een auto of fiets van breedte 0 nodig hebt.
- Voor elk paar = i, j (waar $i < j$), is het mogelijk om te reizen tussen i en j door alleen straten te gebruiken met autostroken met een breedte van minstens $C_{i,j}$. $C_{i,j}$ is het maximale getal met deze eigenschap. Oftewel, voor alle routes tussen locaties i en j is er tenminste één straat met een autostrook met een breedte van maximaal $C_{i,j}$.
- Voor elk paar locaties i, j (waar $i < j$), is het mogelijk om te reizen tussen i en j door alleen gebruik te maken van de straten met fietsstroken met een breedte van minimaal $B_{i,j}$. $B_{i,j}$ is

ook het maximale getal met deze eigenschap.

Kun je de gemeente Lund helpen met het ontwerpen van zo'n stratennetwerk? Omdat er een beperkt budget is, mag je maximaal 2023 straten bouwen. Je kan meerdere straten bouwen tussen dezelfde twee belangrijke locaties, maar je kan geen locatie met zichzelf verbinden. Alle straten kunnen in beide richtingen gebruikt worden.

Input

Op de eerste regel staan twee gehele getallen N en W, het aantal belangrijke locaties in Lund en de breedte van de straten die je kan bouwen.

Op de volgende $N-1$ regels staan de gehele getallen $C_{i,j}$. Op de j-de regel van deze set regels staan alle $C_{i,j}$ met $i < j$. Dus op de eerste regel staat alleen $C_{0,1}$, en op de tweede regel staat $C_{0,2}$ en $C_{1,2}$, op de derde regel staat $C_{0,3}$, $C_{1,3}$, $C_{2,3}$, enzovoort.

Op de volgende $N-1$ regels staan de gehele getallen $B_{i,j}$ op dezelde manier als $C_{i,j}$.

Output

Als het niet mogelijk is om zo'n stratennetwerk te ontwerpen, print 1 regel met "NO".

In andere gevallen, print 1 regel met het gehele getal M, het aantal straten van je netwerk.

Op de volgende M regels, print 3 gehele getallen u, v, b. Dit betekent dat een straat met een fietsstrook van breedte b gaat tussen u en v (en de autostrook heeft dan breedte $W-b$).

Je mag maximaal 2023 straten gebruiken. De straten die je output moeten voldoen aan $0 \leq b \leq W$, $0 \leq u, v \leq N-1$ and $u \neq v$. Je mag meerdere straten gebruiken (mogelijk met verschillende fietsstrook-breedtes) tussen hetzelfde paar van belangrijke locaties.

In het geval van meerdere mogelijk oplossingen, mag je een willekeurige oplossing geven

Constraints and Scoring

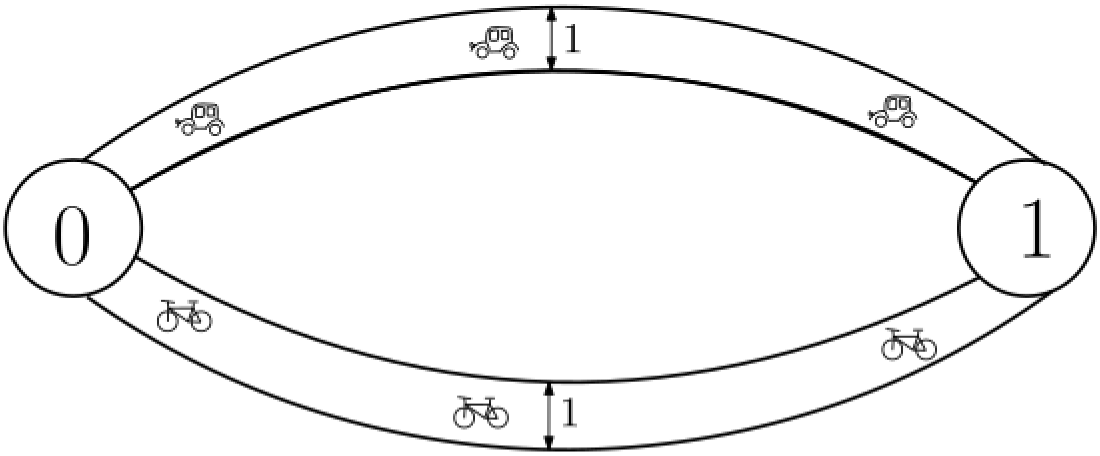
- $2 \leq N \leq 500$.
- $1 \leq W \leq 10^6$.
- $0 \leq C_{i,j}, B_{i,j} \leq W$ for all $0 \leq i < j \leq N-1$.

Your solution will be tested on a set of test groups, each worth a number of points. Each test group contains a set of test cases. To get the points for a test group you need to solve all test cases in the test group.

Group	Score	Constraints
1	10	All $C_{i,j}$ are the same, and all $B_{i,j}$ are the same, $N \leq 40$
2	5	All $C_{i,j}$ are the same, and all $B_{i,j}$ are the same
3	17	$N \leq 40$
4	18	$W = 1$
5	19	All $B_{i,j}$ are the same
6	31	No further constraints

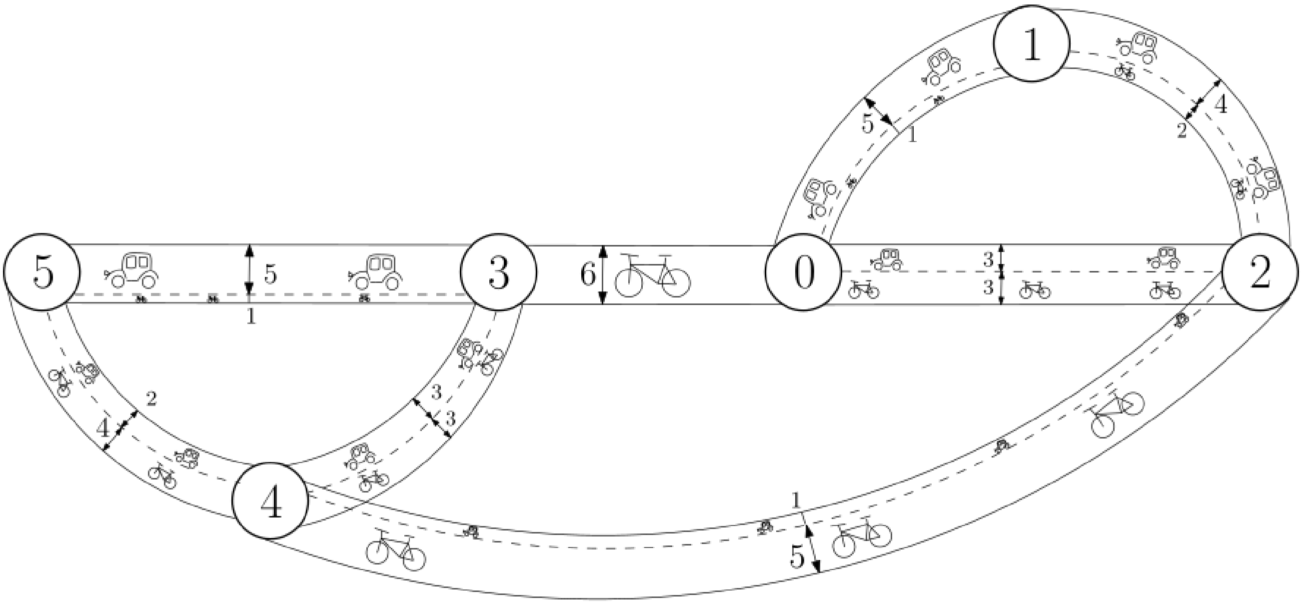
Example

In het eerste voorbeeld, is de breedte van de straat 1 en hebben we een autostrook en een fietsstrook met een breedte van minstens 1 nodig tussen locaties 0 and 1. De oplossing zijn twee straten tussen de locaties, één met een fietsstrook van breedte 1 en één met een autostrook van breedte 1.



In het tweede voorbeeld, is de breedte van de straat opnieuw 1 en er moet tussen elk paar belangrijke locaties een route zijn met een fietsstrook van breedte 1 en er is een route tussen de locaties 1 en 2 en 2 en 3 waar de breedte van de autostrook 1 is voor elke straat. Dit contrasteert met dat als $C_{1,3} = 0$, dat er geen route kan zijn met een autostrook met breedte 1 van 1 naar 3; want we kunnen de twee eerdergenoemde paden gebruiken om zo'n route te maken. Dus het is niet mogelijk om zo'n netwerk te maken.

In het derde voorbeeld, voldoet het stratennetwerk hieronder aan alle condities. Bijvoorbeeld, er is een route met een minimum breedte van de auto-strook $1 = C_{0,5}$ tussen locatie 0 en locatie 5 (bijvoorbeeld door het volgen van de route $0 \to 2 \to 4 \to 5$), een route waarbij de fietsstrook heeft een minimumbreedte $3 = B_{0,5}$ (bijvoorbeeld door het volgen van de route $0 \to 3 \to 4 \to 5$). Tegelijkertijd, zijn er geen routes met een groter minimum voor één van de connecties. Let op, er zijn veel andere oplossingen voor het derde voorbeeld.



0
1
2
3
4
5
6
5
1
5
1
4
2
3
3
4
2
3
3
1
5

Input	Output
2 1 1 1	2 0 1 0 0 1 1
4 1 0 0 1 0 0 1 1 1 1 1 1 1	NO
6 6 5 4 4 1 1 1 1 1 1 3 1 1 1 5 3 2 3 2 6 2 3 3 2 5 3 3 2 4 3 4	8 0 1 1 0 2 3 1 2 2 0 3 6 2 4 5 3 4 3 3 5 1 4 5 4