Problem A. A Story

Input file:	standard input
Output file:	standard output
Balloon Color:	Orange

Once upon a time in a world that looks like ours a story of a person who looks like many of us is about to begin and it unravels like a jigsaw puzzle. Make sure to open your eyes and focus, as things can look like a duplicated version of another person's story but soon you will know that it is exceptionally different than any other story you've experienced.

For it being a great harmonic and very imminent journey where paths may cross. Kudos to those who hone their knowledge and leverage their skills as marching into the future would definitely require them.

Esraa is working as an information security engineer. Every employee in her company has a password which is a permutation of N integers.

All the passwords are stored on the main server. To ensure their safety, Esraa created an encryption algorithm and named it EsraaCipher, the algorithm works in the following way:

1. The algorithm appends a 0 to the end of the permutation.
2. It generates all the rotations (left cyclic shifts) of the permutation.
3. It sorts the generated rotations in lexicographical increasing order.
4. The encrypted password is constructed by taking the last integer of every generated rotation after sorting lexicographically.

Here is an example on how the algorithm works: suppose the permutation $\mathcal{P}$ is {2, 1}, first a 0 is appended to the end of $\mathcal{P}$ so it becomes {2, 1, 0}, now the algorithm will generate all the rotations which are: {2, 1, 0}, {1, 0, 2}, {0, 2, 1}. after sorting the rotations we get: {0, 2, 1}, {1, 0, 2}, {2, 1, 0}. the encrypted password is constructed by taking the last number of every rotation after sorting, so the resulting encrypted password is {1, 2, 0}.

You are hired to try to decrypt and retrieve all the passwords. Can you write a program to decrypt all the $\mathcal{T}$ passwords (given the encrypted password, restore the original password)?

Input

The first line of input contains on integer $\mathcal{T}$ $(1 \leq \mathcal{T} \leq 10^5)$, the number of testcases (passwords to decrypt).

The first line of every testcase contains one integer N $(1 \leq N \leq 10^5)$, the length of the i_{th} password.

The second line of every testcase contains $N + 1$ integers separated by spaces, which represent the encrypted password.

It's guaranteed that the encrypted password contains distinct integers and any integer χ of them in the range $[0, N]$ $(0 \leq \chi \leq N)$.

It's also guaranteed that the sum of lengths of all passwords won't exceed $3\text{x}10^5$

Output

Output $\mathcal{T}$ lines, in the i_{th} line output N integers separated by spaces which represent the i_{th} password after decryption. Note that it must be a permutation of size N.

Example

standard input	standard output
3 4 4 0 1 2 3 2 1 2 0 3 2 3 1 0	1 2 3 4 2 1 3 1 2

Note

Lexicographical order is defined in the following way. When we compare permutations s and t, first we find the leftmost position where $s_i \neq t_i$. If there is no such position (i. e. s is a prefix of t or vice versa) the shortest permutation is less. Otherwise, we compare s_i and t_i.

Problem B. Begins and it

Input file:	standard input
Output file:	standard output
Balloon Color:	White

Just like any story, it begins with an ambitious person, Hisham. And as it moves forward, paths start to appear and connect. And as it takes two to Tango, Endure Capital, the ACPC Community Partner, believing in the relentless execution to build and achieve hyper-growth, starts investing in Hisham's future journey.

Hisham has two integer arrays $\mathcal{A}$ and $\mathcal{B}$, both of length N. He defines the beautiful shifting operations on arrays χ of length N as doing one of two operations:

- Shift all elements to the right, i.e. the new value of χ_i is equal to the old value of χ_{i-1}, and the new value of χ_1 is equal to the old value of χ_N.
- Shift all elements to the left, i.e. the new value of χ_i is equal to the old value of χ_{i+1}, and the new value of χ_N is equal to the old value of χ_1.

Hisham calls index I beautiful if $\mathcal{B}_i$ divides $\mathcal{A}_i$, i.e. ($\mathcal{A}_i \bmod \mathcal{B}_i == 0$). Also he defines the beauty of the array $\mathcal{A}$ as the number of beautiful indices.

You can do the beautiful shifting to either $\mathcal{A}$ or $\mathcal{B}$ as many times as you want (possibly zero). What is the maximum possible beauty of Array $\mathcal{A}$?

It's guaranteed that the elements of the array $\mathcal{B}$ are distinct.

Input

The first line of input contains one integer $\mathcal{T}$ ($1 \le \mathcal{T} \le 100$), the number of testcases.

The first line of every testcase contains one integer N ($1 \le N \le 10^5$), the length of the arrays $\mathcal{A}$, $\mathcal{B}$.

The second line of every testcase contains N integers separated by spaces, which represent the array $\mathcal{A}$ ($1 \le \mathcal{A}_i \le 10^5$).

The third line of every testcase contains N integers separated by spaces, which represent the array $\mathcal{B}$ ($1 \le \mathcal{B}_i \le 10^5$), and all $\mathcal{B}_i$ are distinct.

It's guaranteed that the sum of N over all testcases won't exceed $3x10^5$.

Output

Output $\mathcal{T}$ lines, in the i_{th} line output one integer which represents the maximum possible beauty of arrays $\mathcal{A}$ for the i_{th} testcase.

Example

standard input	standard output
4 3 1 2 4 3 2 1 4 3 1 4 1 5 9 2 6 2 1 2 1 2 1 2 4	2 1 2 0

Problem C. Can look like a

Input file:	standard input
Output file:	standard output
Balloon Color:	Yellow

A deja vu is a French word expressing the feeling that one has lived through the present situation before. It seems that this story of the passionate Hisham can look like a story written by a different self in a different universe, yet something seems to be a bit different.

Hisham is a knight who fights bad monsters. There are N monsters. Each monster has a power. Hisham needs to neutralized all of bad monsters to be the one true knight of the realm.

Hisham can neutralize the bad monsters in any order. In one operation, he can choose a monster that has a positive power, and neutralize that monster. When Hisham neutralizes a monster with power p, all the monsters' powers will decrease by p_i and you earn p points. A monster is said to be neutralized when its power reaches zero or less.

What is the maximum number of points Hisham can earn after neutralizing all the bad monsters?

Input

The first line of input contains one integer T $(1 \leq T \leq 2000)$, the number of testcases.

The first line of each testcase contains one integer N $(1 \leq N \leq 10^5)$, the number of bad monsters.

The second line of each testcase contains N integers p_i $(1 \leq p_i \leq 10^9)$, the power of the i_{th} monster.

It's guaranteed that the sum of N over all testcases won't exceed $3x10^5$.

Output

Output T lines, in the i_{th} line output one integer which represents the maximum number of points you can earn after killing all the monsters of the i_{th} testcase.

Example

standard input	standard output
2 4 1 3 2 2 3 10 6 2	3 10

Problem D. Duplicated version but

Input file:	standard input
Output file:	standard output
Balloon Color:	Silver

A deja vu is a French loanword expressing when one feels they have lived through the same situation in the past. It almost looked like a duplicated version and they certainly got us in the first part, but even when things look exactly similar between Kristin and Kareem, the tiniest butterfly movement can make the whole difference.

Kareem was bored so he decided to draw a tree of N vertices, rooted at vertex 1.

The next day, Ossama came and saw that Kareem's tree needed more beauty, so he decided to make each vertex v assigned with a beauty value $\mathcal{B}_v$.

The day after that, Ossama woke up and took another look at the tree and found out that it still needs some changes, but he did not know what to do, so he gave the task to you.

You will be given the final tree with assigned beauty values, you need to remove some vertices from the tree (possibly all of them) such that the resulting forest of trees has maximum possible total beauty value, but it should be satisfying 2 main conditions:

- For each pair(u, v) of the remaining vertices, it must hold that $dist(u, v)$ is even in the given tree.
- If we insert the remaining vertices in an array and sort it according to their level (depth in the given tree, then for each two adjacent vertices u, v, it must hold that: $(\text{Level}_u = \text{Level}_v)$ or $(\mathcal{L} <= | \text{Level}_u - \text{Level}_v | <= \mathcal{R})$.

Note that:

- the beauty value of a tree is equal to the sum of beauty values of all its vertices.
- the beauty value of a forest is equal to the sum of beauty values of all its trees.

Input

The first line of input contains one integer $\mathcal{T}$ $(1 \leq \mathcal{T} \leq 1000)$, the number of testcases.

The first line of every testcase contains three integers N, $\mathcal{L}$, $\mathcal{R}$ $(1 \leq N \leq 10^5)$, $(1 \leq \mathcal{L} \leq \mathcal{R} \leq 10^9)$, the number of vertices on the tree, the $\mathcal{L}$ and $\mathcal{R}$ values on the second condition.

Then $N - 1$ lines follows. Each line contains two integers u, v $u != v$ separated by spaces, which represents the edges of the tree.

Then one line follows, which contains N integers separated by spaces, which represents the array $\mathcal{B}$ $(-10^9 \leq \mathcal{B}_i \leq 10^9)$.

It's guaranteed that the sum of N over all testcases won't exceed $3\text{x}10^5$.

Output

Output $\mathcal{T}$ lines, in the i_{th} line output one integer which represents the maximum possible total beauty value for the i_{th} testcase.

Example

standard input	standard output
3 9 1 3 7 3 1 6 5 2 9 7 4 9 8 4 3 2 1 8 1 -1 2 -1 0 -1 0 2 2 1 1 3 -1 9 1 3 7 3 1 6 5 2 9 7 4 9 8 4 3 2 1 8 -3 -1 -4 -1 -5 -9 -2 -6 -5	6 0 0

Problem E. Exceptionally different

Input file:	standard input
Output file:	standard output
Balloon Color:	Red

Reading through others' lives, Yosri discovers that everything looks so familiar, yet everything is so exceptionally different. Yosri decides that for it being a great harmonic imminent journey, it is worth a little bit of a spoiler alert. But eventually, kudos to those who figure the story out. And as always Coach Academy, the ACPC Training Partner, jumps into the picture with training opportunities to provide guidance and lend a hand to Yosri through his big journey ahead.

Yosri has an array $\mathcal{A}$ of N elements. He asks you to find the number of XOR-GOOD triples (i, j, k) $(1 \le i < j < k \le N)$.

Yosri calls a triple (i, j, k) XOR-GOOD if and only if: $(\mathcal{A}_i \oplus \mathcal{A}_j \oplus \mathcal{A}_k)$ is a power of 2. where $\oplus$ represents the bitwise XOR function.

Note that, according to the Yosri, 1 **is not actually a power of 2.**

Input

Thee first line of input contains one integer $\mathcal{T}$ $(1 \le \mathcal{T} \le 1000)$, the number of testcases.

The first line of every testcase contains one integer N $(3 \le N \le 2000)$, the length of the array $\mathcal{A}$.

The second line of every testcase contains N integers separated by spaces, which represent the array $\mathcal{A}$ $(1 \le \mathcal{A}_i \le 10^6)$.

It's guaranteed that the sum of N over all testcases won't exceed 2000.

Output

Output $\mathcal{T}$ lines, in the i_{th} line output one integer which represents the number of XOR-GOOD triples for the array $\mathcal{A}$ for the i_{th} testcase.

Example

standard input	standard output
2 7 4 2 1 2 2 5 1 3 2 5 3	8 1

Problem F. For it being a

Input file:	standard input
Output file:	standard output
Balloon Color:	Blue

It is the year 2708, and people are so excited about the newest inventions. Who could imagine that after so many years we are finally here gathering onsite for the annual Earth Collegiate Programming Championship (ECPC). It is the new era of ECPC. For it being so full of potentials the ECPC family can't wait to pursue their goals and future plans, but first they encourage everyone to enjoy the moment.

It's been 1200 years since the beginning. With the advancement of technology, the Grey Time Operation team is having a journey inviting prodigies from the past. But of course they need to identify which era each team member is heading to.

The Grey team knows that:

- 1508 was the new era for "**pocket watches**".
- 1608 was the new era for "**telescopes**".
- 1708 was the new era for "**steam engines**".
- 1808 was the new era for "**telephones**".
- 1908 was the new era for "**airplanes**".
- 2008 was the new era for "**terraforming**".
- 2108 was the new era for "**exploration**".
- 2208 was the new era for "**relaxing**".
- 2308 was the new era for "**going above and beyond**".
- 2408 was the new era for "**second chances**".
- 2508 was the new era for "**the origin**".
- 2608 was the new era for "**empowering the youth**".
- 2708 was the new era for "**ECPC**".

Now, before you begin your celebration with the new Era, we need to make sure that history is well documented. We will ask you about a year from the above list of years, and you need to tell us which era was it. See the examples for more clarity.

Input

You will be given only one integer number Y.

$Y \in \{1508, 1608, 1708, 1808, 1908, 2008, 2108, 2208, 2308, 2408, 2508, 2608, 2708\}$.

Output

Print what is the era of year Y is famous for.

Examples

standard input	standard output
2708	ECPC
2308	going above and beyond
2608	empowering the youth

Problem G. Great

Input file: standard input
Output file: standard output
Balloon Color: Rose

What an eventful year, it's 2709, a year since Fouad started his plan supporting the Interstellar Collegiate Programming Championship (ICPC). A pursuit of one's goal has never been so fulfilling. As the impact is so rewarding. The great plan continues with the great support of the Arab Academy for Science and Technology (AAST), the ACPC Headquarter. It is for sure, a base reason for establishing Fouad's future plans.

Fouad is really busy travelling across the universe to accommodate for his ambitious plans for the ICPC. The universe consists of N galaxies connected by $N - 1$ bidirectional quantum tunnels and every galaxy is connected to all other galaxies directly or indirectly by following some path.

Fouad asks Ali to plan for his visit. Ali always writes a quantum software for this king of tasks but also Ali is sometimes random so he implements his software with a touch of randomness.

Ali's software choose (if possible) an unordered pair of 2 **different** galaxies (u, v) at random with equal probabilities. $\text{Pairs}(u, v)$ and (u, v) as the same pair. Then it marks all the galaxies along the path between u and v including u and v as visited and also it marks all the quantum tunnels where at least one of their end points (galaxies) is on that path as visited as well. All other galaxies and quantum tunnels are marked as unvisited. This way, the unvisited galaxies and quantum tunnels are divided into a number of systems (component) which will be easier to plan for the visits.

Fouad wants to find the expected number of systems (components) after dividing the galaxies and quantum tunnels in the previously described way. Ali is so busy right now, and he asks you to take this task. Can you help him compute this answer?

Input

The first line contains a single integer T $(1 \le T \le 10^5)$, the number of testcases.

The first line of every testcase contains a single integer N ,$(2 \le N \le 10^5)$, the number of galaxies.

Followed by $N - 1$ lines, each line contains two integers u, v which means there is an undirected quantum tunnel connecting galaxies u and v .

Its guaranteed that the galaxies are connected i.e. every galaxy is reachable from any other galaxy, and there are no self loops or multiple edges.

Also it's guaranteed that the sum of N over all test cases won't exceed $3\text{x}10^5$.

Output

Output T lines, the i_{th} line contains one number, the answer for the i_{th} testcase.

Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}.

Example

standard input	standard output
2 3 1 2 1 3 5 1 2 1 3 2 4 2 5	0.666666667 1.700000000

Problem H. Harmonic

Input file: standard input
Output file: standard output
Balloon Color: Black

A journey is never the end. It is always only a checkpoint that was completed if you really think about it. As one path comes to an end, it suddenly forks into many. Abeer was just there, her choice of the next move was everything harmonic for our story.

Abeer has a grid of size 2^N x 2^N. All cells in the grid are empty except for one cell at (X, Y) which has a wall.

Abeer calls a square of size 2 x 2 without one corner cell an L-shaped domino piece. Abeer asks you to fill the grid with $K = \frac{4^N - 1}{3}$ L-shaped domino pieces, so that no two dominos intersects, nor any cell remains empty.

Print any valid way of filling the grid.

It's guaranteed that **at least one such way exists**.

Input

The first line of input contains one integer T $(1 \le T \le 1000)$, the number of testcases.

The first line of each testcase contains one integers $N (1 \le N \le 8)$.

The second line of each testcase contains two integers X, Y $(1 \le X,Y \le 2^N)$, representing the cell with the wall.

It's guaranteed that the sum of 2^N x 2^N over all testcases won't exceed $3 x 10^5$.

Output

For each testcase output 2^N lines. Each line contains 2^N integers, representing the filled grid. For the cell (X, Y) it should contain value 0. For every L-shaped domino piece, its cells should contain the same integer. There should be exactly K L-shaped domino pieces, each piece is represents by an integer from 1 to K.

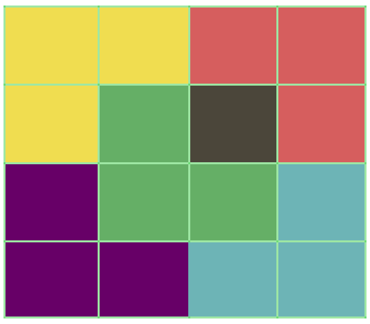
Example

standard input	standard output
1 2 2 3	2 2 3 3 2 1 0 3 4 1 1 5 4 4 5 5

Note

the image below is the answer to the sample, where:

- the black cell (2,3) is the wall.
- the green cells (2,2), (3,2), (3,3) is the piece with id=1 in the output.
- the yellow cells (1,1), (1,2), (2,1) is the piece with id=2 in the output.
- the red cells (1,3), (1,4), (2,4) is the piece with id=3 in the output.
- the purple cells (3,1), (4,1), (4,2) is the piece with id=4 in the output.
- the cyan cells (3,4), (4,3), (4,4) is the piece with id=5 in the output.



Problem I. Imminent

Input file: standard input
Output file: standard output
Balloon Color: Green

With the upcoming plans, Lili knows that crossing paths again with others' stories is imminent. She knows that even if things appear to take the same turns, it is always different when you look up close and invest in seeing. Lili is very considerate to this when she plans their next move.

Lili has two integer arrays A and C, both of length N. Also she has M segments represented by a left and a right boundaries.

Lili asks you to remove a subset S of size s of indices ($i_1, i_2, \ldots, i_s$) only once, such that the following should be satisfied:

- Let the remaining indices be $j_1, j_2, \ldots, j_t$. Values ($A_{j1}, A_{j2}, \ldots, A_{jt}$) must be pair-wise co-prime.
- No two indices (k,l) of the removed indices hold that (k divides l) neither (l divides k).
- No two indices (k,l) of the removed indices are covered by the same segment.
- The cost of this removal is $max(C_{i1}, C_{i2}, \ldots, C_{is})$.

Determine if it's possible to remove some subset; if it is possible, find the minimum possible cost to do it.

Input

The first line contains one integer T ($1 \leq T \leq 10^5$), the number of test-cases. Every test-case consists of the following: The first line of every test-case contains two integers N, M ($1 \leq N \leq 10^5$, $0 \leq M \leq 10^5$), the size of the array A and the number of segments.

The second line of each test-case contains N integers, the i_{th} integer represents A_i ($1 \leq A_i \leq 10^7$).

The third line of each test-case contains N integers, the i_{th} integer represents C_i ($1 \leq C_i \leq 10^7$).

The next M lines, represent the segments (ranges), each line contains two integers L_i , R_i ($1 \leq L_i \leq R_i \leq N$).

It's guaranteed that the sum of N over all test-cases won't exceed $3\text{x}10^5$.

It's guaranteed that the sum of lengths of all segments over all test-cases won't exceed 10^6.

Output

Output T lines, in the i_{th} line if it's possible to remove some subset, output the minimum possible cost to do it, otherwise output -1.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>4
5 0
2 4 8 9 16
1 10 100 1000 10000
5 2
2 3 4 9 8
1 10000 100 100000 1000
4 5
2 3
5 1
2 3 4 9 8
1 10000 100 100000 1000
2 3
5 1
2 3 4 9 8
1 10000 100 100000 1000
4 5</td><td>10000
-1
100000
10000</td></tr>
</table>

Problem J. Journey

Input file: standard input
Output file: standard output
Balloon Color: Purple

Have you noticed yet? It's never been about the end of the journey. There is always what is coming next. It is about what happens throughout the journey with all the paths and turns. It is like an Arena, or to be more specific, a Talents Arena, the place where geeks just like Islam find what they have always been looking for!

Islam found a tree of N nodes rooted at node 1. He wants to process a number of queries Q of two types of queries:

- $1\ \chi\ t_1\ t_2$, query of the first type, a magnet appears in node χ $(1 \le \chi \le N)$, between time interval (inclusive $[t_1, t_2]$ $(1 \le t_1 \le t_2 \le 10^6)$ i.e., it appears at time t_1 and disappears after time t_2. The magnet affects all the nodes in the subtree of node χ (including χ).
- $2\ \chi\ t$ query of the second type, a coin appears at moment t $(1 \le t \le 10^6)$ in node χ $(1 \le \chi \le N)$, and disappears after one moment, if there is any active magnet affecting node χ at that time, this magnet catches this coin. If there are multiple magnets the closer magnet catches it.

For every query of the second type, print the node containing the magnet that catches this coin, or -1 if no magnet catches it.

Note that all first type queries will appear first in the input before any second query appears

Input

The first line of input contains one integer T $(1 \le T \le 200)$, the number of testcases.

The first line of every testcase contains two integers N, Q $(1 \le N, Q \le 10^5)$. The number of vertices of the tree, and the number of queries.

Then $N - 1$ lines follows, each line contains two integers $u, v,$ $(1 \le u, v \le N)(u \ne v)$ separated by spaces, which represent the edges of the tree.

Then Q lines follows, each one represents a query. Each query is either $1\ \chi\ t_1\ t_2$ $(1 \le \chi \le N)$, $(1 \le t_1 \le t_2 \le 10^6)$, or $2\ \chi\ t$ $(1 \le \chi \le N)$, $(1 \le t \le 10^6)$.

It's guaranteed that all first type queries will appear first in the input before any second type query appears.

It's also guaranteed that the sum of N over all testcases won't exceed $5\text{x}10^5$

It's also guaranteed that the sum of Q over all testcases won't exceed $5\text{x}10^5$

Output

For every query of the second type, print the number of the node containing the magnet that catches this coin, or -1 if no magnet catches it.

Example

standard input	standard output
1 8 11 1 2 1 3 3 7 7 8 2 4 2 5 2 6 1 2 2 5 1 1 1 10 1 7 2 10 1 1 11 20 2 4 3 2 5 1 2 8 1 2 7 2 2 6 2 2 2 15 2 1 22	2 1 1 7 2 1 -1