National Taiwan University

2013 NTU Final

A	Razzil Darkbrew, the Alchemist	1 s
B	Strygwyr, the Bloodseeker	5 s
C	Nessaj, the Chaos Knight	3 s
D	Ish'kafel, the Dark Seer	8 s
E	Raigor Stonehoof, the Earthshaker	1 s
F	Darkterror, the Faceless Void	1 s
G	Meepo, the Geomancer	2 s
H	Chen, the Holy Knight	4 s
I	Kael, the Invoker	1 s
J	Jakiro, the Twin Headed Dragon	10 s

December 28, 2013

(This page was intentionally left only almost blank.)

Prologue

Those of you that had arrived at this stage had passed through the trial of the ten games several months ago. LodeRunner, Magicka, MonkeyIslands^{TM}, ... etc, well we hope you had a great time.

Now, you stood before the final act of your ICPC saga, and the direst challenges you'll have ever faced are about to unravel.

Brace yourself!!

- Q: How about MineCraft, Terraria, Skyrim, and all the fun games we've missed?
  A: Be careful what you wish for!

A. Razzil Darkbrew, the Alchemist

time limit: 1 second

The sacred science of Chemistry was a Darkbrew family tradition, but no Darkbrew had ever shown the kind of creativity, ambition, and recklessness of young Razzil. However, when adulthood came calling he pushed aside the family trade to try his hand at manufacturing gold through Alchemy. In an act of audacity befitting his reputation, Razzil announced he would transmute an entire mountain into gold. Following two decades of research and spending and preparation, he failed spectacularly, quickly finding himself imprisoned for the widespread destruction his experiment wrought. Yet Razzil was never one to take a setback lightly, and sought escape to continue his research.

In the eternal pursuit for the formula that would transform mountains and seas into gold, the Alchemist has stumbled upon two mysterious ingredients known as the Brimstone and the Ichor. The two ingredients themselves could not turn rubbles into gold, but when mixed with gold dusts (which Razzil has an infinite amount of supply), Brimstone could act as an agglomerative reagent that produces gold in a 1:1 basis, while Ichor serves as the catalyst that would double the amount of product.

Formally (the accurate depiction of the aforementioned chemical process), in a single brew the Alchemist could cook together a units of Brimstone and b units of Ichor with gold dusts ($a + b > 0$), assuming he has sufficient amount of the ingredients. The process would then produce $1^a \times 2^b$ units of gold.

Note that Razzil's formula are all discrete and there is no guarantee on what will happen if non-integer units of Ichor / Brimstone is used in one batch. Therefore only interger units of Ichor / Brimstone is allowed in the alchemy process. Furthermore, gold produced by this process becomes the so-called "duped-gold", and could not be used as ingredient whatsoever.

Initially, the Alchemist has A units of Brimstone and B units of Ichor, and would like to perform several (possibly none) batches of the aforementioned process. How many distinct final amount of total duped-gold (sum of all batches performed) could he be left with? Note that Razzil does not need to use up all his ingredients.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 100000$) indicating the number of test cases.

Each test case contains two integers A, B ($0 \leq A \leq 10^9$, $0 \leq B \leq 1000$), representing the amount of Brimstone and Ichor Alchemist has.

Output Format

For each test case, output the number of distinct final amount of total duped-gold. Since the answer might be large, output it modulo $1,000,000,007$.

Sample Input

```
4
2 0
1 1
5 14
123456789 789
```

Sample Output

```
3
4
288
999995473
```

B. Strygwyr, the Bloodseeker

time limit: 5 seconds

Strygwyr the Bloodseeker is a ritually sanctioned hunter, Hound of the Flayed Twins, sent down from the mist-shrouded peaks of Xhacatocatl in search of blood. The Flayed Ones require oceanic amounts of blood to keep them sated and placated, therefore Strygwyr goes out in search of carnage. Over the years, he has come to embody the energy of a vicious hound; in battle he is savage as a jackal. Beneath the Mask of the Bloodseeker, in the rush of bloody quenching, it is said that you can sometime see the features of the Flayers taking direct possession of their Hound.

During one hunting session in the sacred Nothl Realm, Strygwyr has marked N targets where he is going to harvest blood from. The dwellings of the N targets lie on a tightly connected network. In fact, their reachability relations form a tournament graph, i.e., for any two places i and j, exactly one of them directly goes to the other.

Strygwyr plans to follow a route that:

- It is a Hamilton path over the N places, i.e., each vertex (target's dwelling) is visited exactly once.
- The route may start and end at any vertex.
- A target that he has not yet visited (and killed) might discover his act, in which case Strygwyr needs to tie up the loose end. Thus he requires at any time (when he is at a particular target's dwelling), the distance to any vertex he has not yet visited can be no more than two edges away.

Please plan a valid route for Strygwyr.

Furthermore, to ensure that emergency plan could be generated in short notice, Q queries of pair (v_1, v_2) will be generated. Let u_1 be the vertex out of the two that comes first in your output visit sequence, and u_2 be the other node, you have to output a path from u_1 to u_2 with no more than two edges.

One thing, though, Strygwyr doesn't really know if the sequence you plan for him is valid or not. As long as for all the Q queries you can come up with a path with distance no larger than 2, the Bloodseeker will consider your answer valid.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 15$) indicating the number of test cases.

Each test case starts with two integers N, M ($2 \leq N \leq 10^5$, $2 \leq M \leq 5 \times 10^8$, $2 \mid M$), with N indicating the number of targets, and M a parameter (used below). Then N lines follow, each with two integers m_i, a_i ($0 \leq m_i, a_i < M$), characterizing the tournament graph. For any two nodes i, j in the graph, there is an edge

$$\begin{aligned} &i \to j \quad \text{if } [2(m_i \cdot j + m_j \cdot i + a_i + a_j) + 1] \cdot \text{sign}(i - j) \bmod 2M < M, \\ &j \to i \quad \text{otherwise.} \end{aligned}$$

After that there is an integer Q ($1 \leq Q \leq 5 \times 10^5$) on a single line, representing the number of queried pairs. Each of the following Q lines consists of two integers v_1, v_2, the queried pair of vertices. Vertices are 0-indexed.

Output Format

For each test case, output on the first line the visit sequence you'd want to report to Strygwyr. Note that this sequence need not be truly valid.

Then the following Q lines answer the Q queries, in the same order. If the two vertices in interest are u_1 and u_2 where u_1 comes first in your visit sequence, output a path from u_1 to u_2 using at most two edges. This path must be valid, with u_1 and u_2 obeying the order in your sequence above.

Do not output ant redundant whitespace in your output paths.

Sample Input

```
2
4 4
0 1
1 2
2 3
3 0
4
0 1
0 2
0 3
1 2
4 4
0 1
1 2
2 3
3 0
1
0 1
```

Sample Output

```
2 3 0 1
0->1
2->3->0
3->0
2->0->1
0 1 2 3
0->1
```

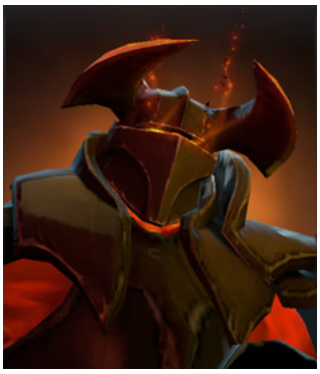
Hint

Note that in the second test case, the output sequence is actually invalid because there is no valid path from vertex 1 to 2. But still, as long as you can answer all the queries successfully according to the sequence you provide, your solution will be judged as correct.

(This page was intentionally left blank.)

C. Nessaj, the Chaos Knight

time limit: 3 seconds



The veteran of countless battles on a thousand worlds, Chaos Knight hails from a far upstream plane where the fundamental laws of the universe have found sentient expression. Upon his steed Armageddon he rides, wading into battle with maniacal frenzy, drawing strength from the disorder of the universe. A physical manifestation of chaos itself, in times of need he calls upon other versions of himself from other planes, and together these dark horsemen ride into battle, as unstoppable as any force of nature. Where rides the Chaos Knight, death soon follows.

During an unexpected shift through time and space, Chaos Knight was teleported into a modern city, Tien-Long. It is a city of efficiency and order, where everything seems to work in their place.

As an agent of entropy, this was by no means a delightful scene for Chaos Knight, and it was not long before he decided to wreak havoc in the city. One of his act was to confuse pedestrians crossing the road.

Pedestrian lights with an alternating periods of green and red signal were employed in Tien-Long to guide the pedistrians crossing the road. Pedestrians have to wait for a green pedestrian light before they could cross a road, and while the light remains green, four "7-segment displays" (an device that display arabic numbers, see below) are used in conjunction to display the time left until signal changes. As you might have expected, when the count down goes to zero, the light singal alters to red, and the count down number disappear. (That is, you will not see "0000" at the end of count down.)

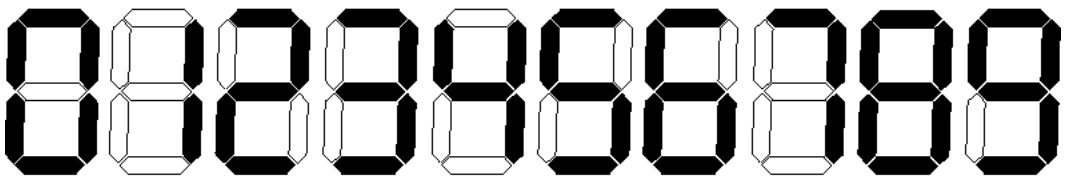


Image: All digits used in 7-segment displays, where black indicates lighted segments.

Chaos Knight's great idea was to secretly sabotage some display segments of each of the "7-segment display", where the broken segments are *exactly the same* for all four displays. The broken segments would never lights and can't be distinguished with a unlit segment. This way, some of the digits could no longer be recognized directly due to the missing display segments. However, note that it is sometimes possible that, if

you observe for a long enough duration, you'd still be able to realize what the digit supposedly displayed is if the 7-segment displays are still intact.

Now, all you know is that Nessaj had broken at most K (out of seven) display segements for some K, and the green light signal counts down from A seconds, but you don't know the exact value of K. You wonder if you walk into the Chaos Knight's piece of work at any time during green light, what is the maximum duration you'll have to observe before you'll be able to deduce the exact time (seconds remaining) when you arrived at the crossing?

Note that the count down doesn't display leading zero. Also note that in this problem, you should treat time as discrete. That is, you always arrive at the pedestrian light at the very beginning of some second, and you observe until the very end of some second.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 20$) indicating the number of test cases.

Each test case contains one integer A ($2 \leq A \leq 9999$) in one line.

75% of the test cases would have $A \leq 999$.

Output Format

For each test case, output 8 space-separated integers, indicating the maximum duration you'll have to observe in seconds when $K = 0, 1, 2, 3, 4, 5, 6, 7$.

Sample Input

```
2
5
14
```

Sample Output

```
1 2 2 2 3 3 4 5
1 2 5 5 5 6 8 14
```

D. Ish'kafel, the Dark Seer

time limit: 8 seconds

Fast when he needs to be, and a cunning strategist, Ish'Kafel the Dark Seer requires no edged weapons to vanquish his enemies, relying instead on the strength of his powerful mind. His talent lies in his ability to maneuver the fight to his advantage.
Hailing from a place he calls "The Land behind the wall," Dark Seer remains an outsider here — a warrior from a realm beyond the veil of this reality.

The most reknowned ability of the Dark Seer is his "Wall of Replica." As it's name suggested, it is a wall of chaotic energy which, when an enemy hero goes across it, an illusion of that hero that fights for Ish'Kafel is generated.

Mathematically, think of the world as a D-dimensional space, and heroes' positions are described by points in the space. The "Wall of Replica" can be considered as a hyperplane, and could be placed at anywhere, along any orientation in the D-dimensional space, except that it *cannot pass through a hero*. Hence, one could imagine that once the wall is placed, all heores on the field are separated into two parts. To maximize the impact on the battle, the Dark Seer has to plan carefully where and how he places the Wall of Replica.

Given the opposing heroes' locations, Ish'Kafel wonders how many different ways are there for him to place the Wall of Replica. Two placements are considered different if there exist two heroes that are on different sides of the wall in one placement, but on the same side in another placement.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 50$) indicating the number of test cases.

Each test case starts with two positive integers N, D ($N \times D \leq 200$), which is the number of enemy heroes and the space dimensionality, respectively. Then follows N lines of D integers, each line representing the coordinate of an enemy hero. All numbers in the input are no more than 10000. Notice that two enemy heroes may have same coordinate.

Output Format

For each test case, output on a single line the number of options Ish'Kafel has. Since the number may be large, output it modulo 1,000,000,009.

Sample Input

```
2
4 2
0 0
0 1
1 0
1 1
3 2
0 0
3 4
5 5
```

Sample Output

```
7
4
```

E. Raigor Stonehoof, the Earthshaker

time limit: 1 second

Like a golem or gargoyle, Earthshaker was one with the earth but now walks freely upon it. Unlike those other entities, he created himself through an act of will, and serves no other master. In restless slumbers, encased in a deep seam of stone, he became aware of the life drifting freely above him. He grew curious. During a season of tremors, the peaks of Nishai shook themselves loose of avalanches, shifting the course of rivers and turning shallow valleys into bottomless chasms. When the land finally ceased quaking, Earthshaker stepped from the settling dust, tossing aside massive boulders as if throwing off a light blanket.

One of the Earthshaker's trademark ability is the spell "Fissure." With such tremendous force, the spell destroys anything within an infinite strip of width W.

In a particular instance, Earthshaker has decided to shatter a gigantic obstacle with his spell Fissure. However, as powerful as the spell Fissure is, it requires an extreme amount of energy. Therefore Raigor Stonehoof has decided that he is only going to cast the spell once.

In order to exterminate such huge obstacle with just one cast of Fissure, the Earthshaker had to do some preparation: He'd cut the obstacle into several chunks prioir to using the spell, and rearrange them so that they could be destroyed by a single Fissure.

Formally, We model the obstacle as a *convex simple polygon* on the plane. Originally, we only have one chunk of obstacle, namely the original convex polygon. In each move, Earthshaker would choose a chunk, and cut it into two by a straight line. Finally when he is done, Raigor Stonehoof would appropiately place the resulting chunks on the plane (possibly with rotation), so that a single Fissure would eliminate them all.

Cutting obstacles cost time as well, though. So the Earthshaker wants to perform as few cuts as possible. Help him figure out what is the minimum number of cuts so a single Fissure is enough to destroy them after reorientation.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 80$) indicating the number of test cases.

Each test case starts with an integer N ($3 \leq N \leq 5000$) and a floating number W ($1 \leq W \leq 10^9$), which is the number of vertices of the obstacle, and the width of "Fissure," respectively. Then N lines follow, where each line contains two integers X_i, Y_i ($-10^9 \leq X_i, Y_i \leq 10^9$), representing the coordinates of vertices of the obstacle. The vertices are given in clockwise order.

It's guaranteed that the answer does not exceed 10, and that changing W by no more than 10^{-5} doesn't change the answer.

Output Format

For each test case, output on a single line the minimum number of cut Earthshaker needs to make.

Sample Input

```
2
4 3
0 0
0 2
3 2
3 0
5 3
0 0
0 3
2 7
5 4
5 0
```

Sample Output

```
0
1
```

F. Darkterror, the Faceless Void

time limit: 1 second

Darkterror the Faceless Void is a visitor from Claszureme, a realm outside of time. It remains a mystery why this being from another dimension believes the struggle for the Nemesis Stones is worth entering our physical plane, but apparently an upset in the balance of power in this world has repercussions in adjacent dimensions. Time means nothing to Darkterror, except as a way to thwart his foes and aid his allies. His long-view of the cosmos has given him a remote, disconnected quality, although in battle he is quite capable of making it personal.

Descent from a universe unknown to human being, Darkterror possesses abilities that even the brightest mind of science cannot understand. Perhaps the flow of matter was beyond our description in the parallel world people have yet to discover, or the continuity of time and space simply does not apply to the Faceless Void itself. Tormented by the hunger of deeper understanding to his strength, Darkterror proceeds to pursue the final piece of knowledge that would make his power whole — the destiny of all universes.

It has been rumoured that the destiny of a universe was written on a scroll. Not much was known about why or who wrote the scroll; all that was known is that the original scroll had been denied from the access of mere mortals, kept in the custody of the Durmand Priory.

With his ability of riftwalk between different dimensions, Darkterror managed to steal a transcript of the scroll from the Priory. However, cautious as the Prioiry was, great efforts were spent to prevent any possible leak of the fate of the universe, for even a trace of revelation would lead to decades-long turmoil. The Prioiry made two copies of the original scroll, inserted bogus characters and permuted them in special ways.

Fortunately for Darkterror, he overheard how the scrolls could be decrypted. The writings on the two copied scrolls could be considered as two long strings consisting of lowercase alphabets, S_1 and S_2. A string T_1 is "reachable" from S_1 if, starting from S_1, it could be obtained after a finite number of swaps of two characters exactly d_1 units away in position. Similarly, A string T_2 is "reachable" from S_2 if it could be obtained after a finite number of swaps of characters d_2 units away. A string P is "contained" if it is simultaneously a prefix of some string T_1^* reachable from S_1 and a prefix of some T_2^* reachable from S_2. The writing on the original scroll is precisely a longest string that is "contained."

Help Darkterror figure out the length of the original writing.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 50$) indicating the number of test cases.

Each test case starts with two integers d_1 and d_2 ($1 \leq d_1, d_2 \leq 200$). The following two lines contain two strings S_1, S_2. Both string would be non-empty, containing lowercase alphabets only. The length of each string would not exceed 200.

Output Format

For each test case, output on a single line the length of the original writing. If there is no possible original writing, output 0 instead.

Sample Input

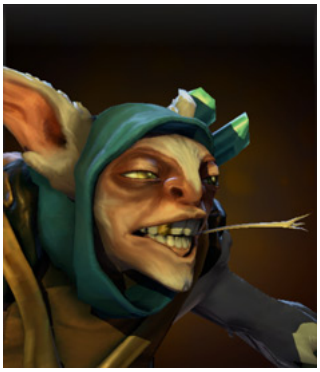
```
3
2 1
tmt
tpm
4 4
abababababab
aaaaaaaaaaaa
1 1
destiny
density
```

Sample Output

```
2
1
7
```

G. Meepo, the Geomancer

time limit: 2 seconds



If you ask me, life is all about who you know and what you can find. When you live up in the Riftshadow Ruins, just finding food can be tough. So you need to cut corners, you need to scrounge, you need to know your strengths. Some of the beasts up there can kill you, so you need a way to trap the weak and duck the strong. On the upside, the ruins have history, and history is worth a lot to some people. There used to be a palace there, where they had all these dark rituals. Bad stuff. If you survived the ceremony, they would shatter a crystal and split your soul into pieces.

Meepo is a mysterious creature living in the Riftshadow Ruins, known for his annoying ability to create replicas of himself to fight alongside him. These ruthless wretched clones together will swarm and crush any unexpected enemy of Meepo caught off guard.

However, these replicas are imperfect and may have different power. For all replicas there are K power traits, and each replica may have varying distribution of strength in these traits. Specificially, for the i-th replica of Meepo, a K-dimensional vector V_i describes its strength in each trait.

Replicas sometimes duel each other. When two replicas i and j with strength V_i and V_j fight each other, the duel which consists of K rounds will go as described below:

Before the duel the two replicas randomly shuffles their strength vectors, and at round t of the duel $V_i[t]$ and $V_j[t]$ are used to determine the winner of the round. If $V_i[t] > V_j[t]$, replica i wins; if $V_i[t] < V_j[t]$, replica j wins. All strength values in all trait vectors are pairwisely distinct so there is always a winner for a single round. After finishing all K rounds, the replica that wins more single rounds prevails. In the case of two replicas winning the same number of rounds, one of the replicas would prevail randomly.

The replicas are merciless and brutal, but also cautious. Whenever one of Meepo's replica finds out that it can 100% outduel another replica, it will proceed to slaughter that replica just for entertainment. Meepo, of course, is not happy about such phenomenon, and would like to prevent his gank squad from killing each other.

Given the strength vectors for all Meepo's replicas, help him determine: what is the maximum number of replicas Meepo can keep in a squad, so that none of them will attempt to kill each other?

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 50$) indicating the number of test cases.

Each test case starts with two integers N and K ($1 \leq N, K \leq 300$), indicating the number of Meepo's replicas and the number of strengh traits, respectively. Each of the next N lines contains K integers (value $\in [0, 10^9]$) in a row, describing the replicas' strength vectors. It is guaranteed that all of the $N \times K$ values in the strength vectors are distinct.

Output Format

For each test case, output in a single line the maximum number of replicas Meepo can bring with him without dispute.

Sample Input

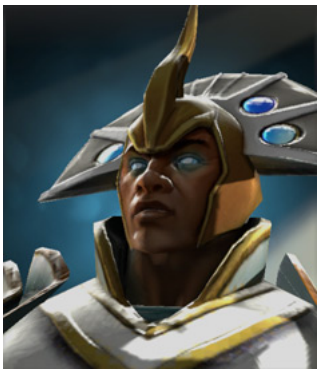
```
2
3 1
3
1
4
2 2
1 2
3 4
```

Sample Output

```
1
1
```

H. Chen, the Holy Knight

time limit: 4 seconds



Born in the godless Hazhadal Barrens, Chen came of age among the outlaw tribes who eked out an existence in the shimmering heat of the desert. Using an ancient form of animal enthrallment, Chen's people husbanded the hardy desert locuthi, a stunted species of burrowing dragon that melted desert sands into tubes of glass where twice-a-year rains collected. Always on the edge of starvation and thirst, fighting amongst their neighbors and each other, Chen's clan made the mistake, one fateful day, of ambushing the wrong caravan.

The Holy Knights are known for their ability to control creeps (animals that dwell in the forests). It is all thanks to their unique spell "Holy Persuasion," which turns neutral creeps into tamed creeps controlled by the holy knights.

Over the years, the holy knights have developed an advanced version of this spell, which affects multiple targeted creeps at once, at the cost of also turning tamed creeps into neutral ones and several limitations detailed below.

There are N creeps in the jungle with some parent-child relationships. The creeps do not obey the normal rule humans are familiar with. Each creep can have any number of parents (even more than two!), and a creep need not have fixed spouse. Still, there would be no cycle in these relationships, and no one would be parent of itself. In fact if you view the creeps as vertices, and parent-child relationships as directed edges, the graph would be a *directed acyclic graph*.

Each of the creeps is either neutral or tamed. Each time a Holy Knight cast "Holy Persuation" on the creeps, he must first choose a *neutral* creep x, and a set of descendants S_x of x (either direct or indirect, either tamed or neutral). Then, the skill would "toggle" the status of each creep in $\{x\} \cup S_x$ (i.e. neutral creeps become tamed, and vise versa). The limitation comes that the total number of creep affected in one cast has to be a multiple of k. Of course, each creep can be affected at most once in one cast of "Holy Persuation".

One day, two Holy Knights are praticing this new spell. They got bored quickly so they came up with a competiton. They decided to cast the spell "Holy Persuation" alternatively on a group of creeps, and the one who no longer has any valid cast loses the competition.

Given the parent-child replationships of these creeps, and the original status (neutral/tamed) of each creep, can you figure out which Holy Knight would have a winning strategy? And if the first Holy Knight has a winning strategy, which creeps can he choose as x in the first move and still guarantee his winning?

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 10$) indicating the number of test cases.

Each test case starts with three integers N, M, k ($1 \leq N \leq 50000$, $0 \leq M \leq 200000$, $1 \leq k \leq 500$), indicating the number of creeps, the number of parent-child relationships between them, and the k in the limitation.

Then follows a line with N integers, each is either 0 or 1. The i-th integer is 0 if the i-th creep is neutral, and is 1 if the i-th creep is tamed.

Finally M lines follow, each with two integers v_i, u_i ($1 \leq v_i < u_i \leq N$), indicating that the v_i-th creep is parent of the u_i-th creep. Each pair of (v_i, u_i) appears at most once.

Output Format

For each test case, output two lines.

In the first line, output "First" if the Holy Knight who cast spell first would win the competition, or "Second" otherwise.

In the second line, output the index of all possible choices of creep x in the first cast of the first Holy Knight, that would still guarantee his winning, separated by spaces. If the second Holy Kinght has a winning strategy, output an empty line.

Sample Input

```
4
3 2 1
0 0 0
1 2
1 3
5 4 2
1 0 0 0 0
1 2
1 3
2 4
3 5
5 5 2
1 0 0 0 0
1 2
1 3
2 4
3 4
3 5
4 3 2
0 0 0 0
1 4
2 4
3 4
```

Sample Output

```
First
1
Second

Second

First
1 2 3
```

(This page was intentionally left blank.)

I. Kael, the Invoker

time limit: 1 second

In its earliest, and some would say most potent form, magic was primarily the art of memory. All the trappings of ritual were merely mnemonic devices, meant to allow the practitioner to recall in rich detail the specific mental formulae that unlocked a spell's power. The greatest mages in those days were the ones blessed with the greatest memories. The most devoted might hope in a lifetime to have adequate recollection of three spells–four at most. But among these early practitioners there was one exception, a genius of vast intellect and prodigious memory who came to be known as the Invoker.

Kael has his distinguished way of combat. Unlike other wizard-craft built on the base of separate spells, the Invoker establishes his power upon elements such as frost, flame, and spark. Therefore every spell of Kael is a series of elements summoned subsequently, and the casting of sophisticated spells could get seriously complicated due to the amount of elements required to summon it.

Formally, elements could be represented by characters from 'a' to 'z'. A spell consists of a series of ordered elements, so it could be considered as a string of lowercase alphabets. Summoning such a long sequence of elements, however, is extermely unrealistic in combat, so Kael has mastered a special form of maneuver to help him cast spells more efficiently.

Suppose the Invoker has summoned a sequence of elements already, seen as an alphabetic string. Instead of creating new elements, Kael could copy a prefix of the existing string, and append a duplicate at the end of current spell.

Should you remember Kael'van, an appretice from Magicka, has been learning from the Invoker Kael ever since the defeat to Daak'Phi. Kael'van was observing how his new master magically maneuver the elements, but often lost track completely. Help him on the task of learning from the Invoker.

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 20$) indicating the number of test cases.

Each test case starts with two integers N, Q ($1 \leq N \leq 1000000$, $0 \leq Q \leq 30000$), indicating the length of current existing spell, and the number of events to be processed, respectively.

Then follows a line with a string of length N, representing the original spell. The string would only consist of lowercase alphabets.

After that Q lines follow. Each line consists of two integer t, k ($t \in \{1, 2\}$, $1 \leq k \leq 10^{18}$), representing a event given in temporal order:

1. If $t = 1$, the Invoker copies a prefix of length k and append it to the end of the current spell.
2. If $t = 2$, Kael'Van wants to know what the k-th element in the current spell is.

It's guaranteed that in both cases, k does not exceed the length of the current spell.

Output Format

For each test case, output a line containing all answers for events with $t = 2$, in the same order as the events.

Sample Input

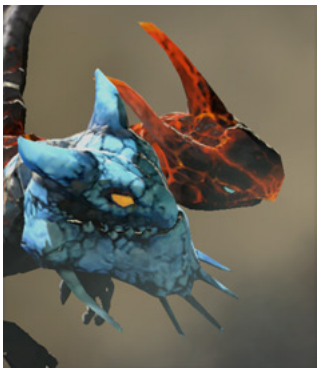
```
2
3 5
mtt
1 2
2 2
2 4
1 5
2 10
3 6
yaw
1 3
1 6
1 12
2 24
2 23
2 22
```

Sample Output

```
tmt
way
```

J. Jakiro, the Twin Headed Dragon

time limit: 10 seconds



Even among magical beasts, a twin-headed dragon is a freak. Equal parts ice and fire, cunning and rage, the creature known as Jakiro glides over charred ice-bound battlefields, laying waste to any who dare stand against it.
Pyrexae dragon clutches always contain two fledglings. Famous for their viciousness even from the first moments of life, newly hatched dragons of this species will try to kill their newborn sibling, and only the strongest survive. By some accident of nature, the freak Jakiro hatched from a single egg, combining in a single individual the full range of abilities within the Pyrexae species, and the powers of ice and fire combine.

With both the mind of cold cruelity and fiery rage fused in one body, it happens rather often the two voices of the twin-headed dragon disagree with each other. In such case a coin is flipped randomly (imaginary, of course!), and one of the two heads' choice of actions is selected accordingly.

In a particular scenario, a series of decisions (with binary outcome) are to be made, and each final decision comes with a particular consequence. Formally, the whole event development could be considered as a binary decision tree T_D, where initially, Jakiro is at the root of the decision tree, and at each branch he chooses one of the two routes, until finally he arrives at some leaf node. Each leaf node represents a final event with unique ID from 0 to $N-1$, where N is the number of leaf nodes.

In order to evaluate the variance between the outcomes that stem from different choice of actions, a tree topolgy is used to define the distance metrics between the N final events. Formally, Given an undirected weighted tree T_C with N nodes numbered from 0 to $N-1$, the distance $D(i,j)$ between final event i and final event j is the distance of node i and node j on T_C.

We are now equipped to define the "decision impact" of a node on the decision tree. For a node v on T_D, one choice could (finally) lead to possible outcome set S_1, while the other could lead to possible outcome set S_2, then the decision impact value $I(v)$ is defined as the expected value of $D(i,j)$, with i an outcome chosen *uniformly* randomly from S_1, and j an outcome chosen *uniformly* randomly from S_2.

Given T_D and T_C, Jakiro wonders, for each node v on decision tree T_D, what is the decision impact value $I(v)$?

Input Format

The first line of the input file contains an integer T ($1 \leq T \leq 40$) indicating the number of test cases.

Each test case starts with an integer N ($2 \leq N \leq 100000$), indicating the number of final events. Starting from the second line of each test case is the description of T_D given in parentheses form, i.e.

- ID := $0 \mid 1 \mid 2 \mid \cdots \mid N-1$
- Tree := ID $\mid$ ($\text{Tree}_{\text{left}}$, $\text{Tree}_{\text{right}}$)

After that comes the description of T_C, which is simply $N-1$ lines, each with three integeres v_i, u_i, l_i ($0 \leq v_i, u_i < N$, $1 \leq l_i \leq 1000$), representing an edge between v_i and u_i with length l_i in T_C.

Output Format

For each test case, output $N-1$ lines where each line consists of a single simplified fraction. The number on line i represents the decision impact value of the i-th visited non-leaf node in T_D's *pre-order* traversal.

Sample Input

```
2
4
((0,1),(2,3))
0 2 1
1 2 1
2 3 1
4
((0,(1,2)),3)
0 2 1
1 2 1
2 3 1
```

Sample Output

```
3/2
2/1
1/1
5/3
3/2
1/1
```