Problem A – Arrows and Quiver

Arjuna has a test in archery from his guru, Dronacharya. He carries with him a bag full of N arrows, each arrow of a specific (distinct) kind. The test is designed as a sequence of M targets. Each target can be hit by only a specific kind of arrow. It is a common practice to hold a small number of arrows in quiver on one's back because it saves you from the trouble of carrying a big bag of arrows with you everywhere. Every quiver has a fixed capacity.

Being such an amazing archer, he takes no time to fire an arrow from his quiver and hit his target. He also takes no time to retrieve his arrow from the target for use in possible future targets. But his quiver size is not very large. And since his bag of arrows is so large, he incurs a cost of 1 time unit whenever he needs to search his bag and load the respective arrow into his quiver.

He uses the following scheme for filling his quiver. Initially, it is empty.
a) If the arrow required to shoot is present in the quiver, then his firing is instant. After firing the arrow, he obviously gets it back and puts it in his quiver, all this in no time.
b) If the arrow required to shoot is not present in the quiver, he takes one time unit to find it, shoot it, and put it back in the quiver.
c) If his quiver cannot hold the arrow (i.e., the quiver is full to its maximum capacity), he takes the least recently used arrow and shoves it back into his bag instantly; and then places the arrow that he just shot in the quiver.

Now, he needs to get done with this test quickly. In particular, he has a time limit of C before which he has to go meet his lover, Subhadra. Given that he has N arrows and M targets (which he has to fire in order), find the minimum capacity of the quiver S needed in order to finish his test by time C.

Since Dronacharya does not approve of his love, it might be impossible for him to finish in time. In which case, output -1.

Input:
The first line contains T, the number of test cases. The description of T Test cases follow. The first line of each test case contains 3 space separated integers N, M and C denoting the total number of arrows, the count of targets that Arjuna has to fire at and the maximum total time he can afford to spend on the test respectively. The next line of each test case contains M space separated integers (each between 1 and N), denoting the order of arrows which he needs to fire at the targets.

Output:

For each test case, output a single integer denoting the minimum quiver size S, such that the total time to finish the test doesn't exceed C. If it is not possible to finish in time, output -1.

Constraints:

T <= 10
1 <= N, M <= 10^5
1 <= C <= M

Example:

Sample Input

```
3
5 8 5
1 2 3 4 5 3 4 5
3 3 1
1 1 2
4 5 2
1 2 1 2 1
```

Sample Output

```
3
-1
2
```

Time Limit: 3 sec
Memory Limit: 256 MB

Problem B – Sorted Queues

The Pandavas are coming back to Hastinapura after winning the war with the Kauravas. Yaaay!!! The people of Hastinapura want to welcome them with a long procession. They decide to stand in two parallel lines A and B of size N each. (In the days of our glorious past, Indians knew how to stand in orderly lines.) A[i] is the height of i-th person in line A and B[i] is the height of i-th person in line B.

The people want to arrange themselves in strictly increasing order of their heights so that they can all see the dignitaries on stage in front. To avoid the inevitable chaos if everyone started to move around, they are allowed to do only one kind of operation:
* pick an index i, and swap the i-th person from line A with the i-th person from line B (swap A[i] and B[i]).

We won't ask you whether you yourself are well-behaved when standing in a line, but you should surely be able to determine
a) whether they can perform swaps so that in the end, the people in A and B have their heights sorted in strictly increasing order.
b) and if (a) is possible, the minimum number of steps to do it, with each swap being counted as a step.

Input:
The first line contains T, the number of test cases. The description of T test cases follow.
The first line of each test cases contains integer N, the size of arrays A and B.
The next line contains N space separated integers, which represent the heights of people in line A. The third line contains N space separated integers which represent the heights of people in line B.

Output:
For each test print a single integer which is the minimum number of steps required to make both A and B sorted. Print '-1' if they cannot be put in sorted order.

Constraints:
1 <= T <= 100
1 <= N <= 10000
0 <= A[i], B[i] <= 10^9

Sample Input:

```
3
4
1 4 4 9
2 3 5 10
3
1 2 3
3 2 1
4
2 1 6 5
0 3 4 7
```

Sample Output:

```
1
-1
2
```

Time Limit: 2 sec
Memory Limit: 256 MB

Problem C – Ganga Fort

The river Ganga is known to be long - as long as N kilometers. The Kauravas have set up M fortifications on its northern bank. The ith fortification goes from kilometer A[i] to kilometer B[i](both inclusive) and takes C[i] units of ammunition to destroy. That is, after destroying the fortification, your ammunition store decreases by C[i]. The Pandavas wish to cross the river from the south with their army, which is K kilometers wide. Being K km wide, some parts of the army may have a clear path to the other (northern) side of the river, and others may encounter one or more fortifications that need to be destroyed; but for the whole army to cross, any and all fortifications in their way should be destroyed completely. Also, they do not wish to split up the army as they cross, since the Kauravas could ambush them there.

What is the minimum total units of ammunition they need at the beginning to overcome enough fortifications to let their army cross? The Pandavas can choose any contiguous segment of river of length K and decide to cross their army in that segment, destroying any fortifications in their way.

Input:

The first line contains the number of test cases T. T test cases follow. For each test case, the first line contains integers N, M and K. The next M lines describe the fortifications. The ith line contains A[i], B[i] and C[i].

Output:

Output T lines, containing answers for the respective cases.

Constraints:

1 <= T <= 100
1 <= K <= N <= 10^9
1 <= M <= 10^5
1 <= A[i] <= B[i] <= N
1 <= C[i] <= 1000

Sum of M over all test cases in a single file does not exceed 5*10^5 .

Sample Input:

```
3
10 3 10
1 5 20
2 7 30
3 8 40
10 3 5
1 3 2
1 4 2
7 10 5
10 3 3
1 3 1
7 8 1
9 10 1
```

Sample Output:

```
90
4
0
```

Explanation:

For the first test case, you need the entire length of Ganga to cross. The Pandavas have to destroy all the fortifications.
For the second test case, you can destroy fortifications 1 and 2 to cross.
For the third test case, you can use the segment 4-6 of the Ganga. There is no need to destroy any fortification.

Time Limit: 3 sec
Memory Limit: 256 MB

Problem D – Draupadi's Swayamvar

It is Draupadi's Swayamvar and this time they choose the following rules for the competition to win her hand in marriage. The Pandavas (team A) and the Kauravas (team B) each come with N members to the event. Also, you know for each person in the two teams, who would win in a battle between them (i.e. you know whether person i of team A would win against person j of team B for all i and j). The rules are as follows:

1. The captain of the Kauravas (team B) chooses some subset of people from the Pandavas (team A). Let the number of people chosen be K (1 <= K <= N).
2. The captain of the Pandavas then chooses K people from the Kauravas team for the contest. He knows which K people were chosen in step 1 before he makes his choice.
3. Next, individual fights take place. In each fight, first a member of team B comes up to the arena. Then, some member of team A challenges him and the fight starts. The winner survives and can take part in subsequent fights. The loser is eliminated.
4. Once all fights are done (when all chosen members of one of the teams have been eliminated), the team with any surviving chosen members wins.

Considering that all decisions (to choose the initial set of K members, and to choose fighters for individual fights) are made optimally by both sides, which team wins?

If the Kauravas win, also output the lexicographically smallest set of members among the Pandavas that should be chosen for the fight in step 1. The following is an example of lexicographic ordering: If team A has 3 members {1, 2, 3}, then the lexicographic ordering of (all) subsets of team A would be {}, {1}, {1, 2}, {1, 2, 3}, {1, 3}, {2}, {2, 3}, {3}.

Input:

The first line contains T, the number of test cases. The description of T test cases follow. The first line of each test case contains a single integer N, denoting the number of people in each of the teams. Each of the following N lines contain a string containing N characters each. Each character is either '0' or '1'. The jth character in the ith string is '1' iff person i from team A would win in a fight with person j from team B.

Output:

For each test case, output a single line containing the desired answer for the test case. If team B wins, then output the indices (1 based) of the players chosen from team A in step 1 (space separated). If there are multiple solutions, then output the lexicographically smallest solution. Instead if team A wins, output -1.

Constraints:

1 <= T <= 100
1 <= N <= 50
All strings contain only the characters '0' or '1'.

Sample Input

```
3
2
01
01
3
101
110
011
3
111
000
111
```

Sample Output

```
1 2
-1
2
```

Time Limit: 5 sec
Memory Limit: 256 MB

Problem E – Army Formation

A key element of war strategy is the arrangement of your armies on the battlefield. Each soldier in the Pandava army specializes in exactly one of M possible specializations (archery/infantry/cavalry to name a few). They have decided to choose a 2-rows N-columns (2xN) formation for their army. Now, it is a good thing to ensure that each person's specialization complements those in his row and his column. They wish their army formation to have no two soldiers in the same row or same column having the same specialization. Can you help them by finding the number of ways in which this can be achieved?

Input:
The first line contains T, the number of test cases. The description of T test cases follow. Each test is described by a single line containing 2 space separated integers, denoting N and M.

Output:
For each test case, output a single line containing the number of ways in which the Pandavas can form their army. Since the numbers may be large, output all answers modulo 1000000007.

Constraints:
T <= 1000
1 <= N, M <= 1000

Sample Input
3
1 1
2 2
2 3

Sample Output
0
2
18

Time Limit: 2 sec
Memory Limit: 256 MB

Problem F – Cheap Gold

Yudhishtira was a rich king, but was also frugal, and when he built his grand golden palace at Indraprastha, he found that he was running over budget. Draupadi wanted too many dressing rooms, Bhima wanted a big workout room, and Arjuna had so many divine benefactors who presented him with weapons that he needed many rooms to store them. So he decided to send emissaries to the south to shop for gold, as he knew that gold prices at different cities along the road to the south were often low.

The big road to the south of the empire had a total of C cities along it, and the gold price at i-th city was p[i], where 1<=i<=C.

A number of shoppers, N to be precise, decided to each go along the road; the k-th shopper (1<=k<=N) would visit the first s[k] consecutive cities, and buy gold at the cheapest price he can find among the first s[k] cities. After coming back, all shoppers will sell the gold at the price at which they bought the gold (too fierce competition).

Yudhishtira can choose to buy the needed gold from any one of the shoppers. Given the list of prices p[i] and the number of cities each shopper visited s[k], find the lowest price that Yudhishtira had to pay for the gold.

Input:

The first line contains C, the total number of cities along the road.
The second line contains C space separated integers, the prices at each city p[1], p[2], ..., p[C].
The third line contains N, the number of shoppers.
The fourth line contains N space separated integers, the number of cities visited by each shopper, s[1], s[2], …, s[N].

Output:

The output should have a single integer, the lowest price at which Yudhishtira can buy the gold.

Constraints:

1 <= C <= 10000
1 <= N <= 10000
0 <= p[i] <= 10^6
1 <= s[k] <= C

Sample Input:

Problem F – Cheap Gold

5
3 4 1 5 2
3
1 3 2

Sample Output:

1

Explanation:

There are 5 cities.

The first city has price 3, the second city has price 4, so on. There are 3 shoppers who visit 1, 3 and 2 cities respectively. The lowest price each encounters is 3, 1, and 4 respectively. So the cheapest gold is purchased by the second shopper, at the price of 1.

Time Limit : 2 sec
Memory Limit: 256 MB

Problem G – Bridges

Duryodhana built an archipelago (collection of islands) with P islands in the holy lake of Mansarovar. Each island is a narrow rectangle running North-South, with the i^th island having length k[i].

The islands are numbered such that the westernmost (leftmost) island is number 1, next is 2 ... , with the easternmost (rightmost) being P. The southernmost point of ith (1<=i<=P) island is labeled (i, 1) and northernmost point is (i, k[i]). The points in between are labeled according to their distance from the southern-most point.

A walk-a-thon is being organized with K Kaurava walkers, where each walker is labeled uniquely with labels from 1 to i. Every walker starts on island 1 and ends on island P.
The i^th walker starts at (1, i) and ends at (P, i).
All walkers start simultaneously from island 1 and have to reach island P.

Bridges have been built to enable walkers to move between islands.
A bridge connects two adjacent islands and permits traffic only from west to east. A bridge starts at point (i, a) on island i and ends at point (i+1, b) on island i+1.
No two bridges cross, but more than one bridge can start/end at the same point.
In a single time unit, all walkers move to the next island, i.e., if they are at island i currently, they need to move to island i+1 at the next point of time. The walkers move from one island to another by walking along the bridges.

As soon as a walker reaches the end of one bridge, he or she will immediately take another bridge starting at the same point and leading to the next island, i.e., the walker is not allowed to move within the island and is not allowed to take rest.

A walk-a-thon is said to be valid if two walkers never start at or arrive at the same point of any island at the same time. Two walk-a-thons of K walkers through the archipelago are considered different if at least one walker took a different route. Two walk-a-thons that have the same routes but differ only in the identity of the walkers are considered identical.

Assuming all walkers start at the same time, find the number of valid and different walk-a-thons that can take place.

Input:
The first line contains T, the number of test cases. The description of T test cases follow.
The first line of each test case contains three integers, P, K and B denoting the number of islands, the number of walkers, and the number of bridges respectively.
The next line contains P integers, the ith integer represents k[i], the length of ith island.
The length of first and last island will always be K.

The starting points of ith walker is (1, i), and his ending point is (P, i).
The next B lines contain three integers each: l, a, b which indicates a bridge from (l, a) to (l+1, b)

Output:
For each test case, output a single integer, the number of walk-a-thons possible modulo 10^9+7.

Sample Input
```
2
3 3 6
3 3 3
1 1 1
1 2 2
1 3 3
2 1 1
2 2 2
2 3 3
3 3 14
3 5 3
1 1 1
1 1 2
1 2 2
1 2 3
1 2 4
1 3 4
1 3 5
2 1 1
2 2 1
2 3 1
2 3 2
2 4 2
2 4 3
2 5 3
```

Sample Output
```
1
6
```

Constraints:
T <= 8
1 <= P <= 80
1 <= K <= 50
0 <= B <= 8000
1 <= k[i] <= 80
The bridges between two adjacent islands do not intersect, except at end points.
k[1] = k[P] = K
1 <= L < P
1 <= a <= k[L]
1 <= b <= k[L+1]

Time Limit: 5 sec

Memory Limit: 256 MB

Problem H – House of Lac

Lakshagraha was a house built of lacquer, made by the Kauravas to kill the Pandavas. The Kauravas wanted to burn the house down when the Pandavas were asleep at night. But poor Kauravas -- once again they underestimated their cousins. Having been warned of the nefarious plan, the Pandavas had had an underground set of passages built for escape.

The underground rooms and passages were in the form of a n*m grid where every cell is either free or blocked by a pillar. The Pandavas start at a free cell and they need to reach a destination cell(which can be free or blocked).

The following are the allowed valid moves:

* Move from an empty cell to another adjacent empty cell. (Cells sharing a common side are considered adjacent).
* If an adjacent cell is blocked, then set the edge of the blocked cell on fire.
* If two or more (distinct) edges of a blocked cell are set on fire, then the blocking pillar burns down and clears the cell. After the fire, the cell becomes empty.

Initially no edge of any blocked cell is set on fire. Help the Pandavas find whether it is possible to reach the destination (target cell), because they are the good guys.

Input:

The first line contains T, the number of test cases. The description of T Test cases follow. The first line of each test case consists of 2 space separated integers n and m, denoting the dimensions of the grid (n x m grid). Each of the following n lines contain m characters each, where the jth character of the ith line denotes the state of the cell located at the jth column of the ith row of the grid. Each cell can either be blocked (denoted by '*'), or free (denoted by '.'). The next line of each test case consists of 4 space separated integers sx, sy, ex, ey, where (sx, sy) denotes the cell where you are initially located at, and (ex, ey) denotes the destination cell (1 based indices).

Output:

For each test case, output a single line containing "YES" or "NO" (quotes for clarity), denoting whether it is possible to reach the destination cell from the given starting cell by making valid moves as described above.

Constraints:

T <= 20
1 <= n, m <= 500
1 <= sx, ex <= n
1 <= sy, ey <= m
The starting cell is always empty.

Sample Input

```
3
2 3
.*.
...
1 1 1 3
3 3
..*
..*
.*.
2 1 3 3
2 3
.*.
**.
2 3 1 2
```

Sample Output

```
YES
YES
NO
```

Time Limit: 2 sec
Memory Limit: 256 MB

Problem I – Death Row

Yudhishthira and his family had given up their kingdom after many years of peaceful rule, and were trekking in the Himalayas. They were going in single file up a narrow mountain road, with Yudhishtira in the lead. As the trek progressed, Yudhishtara noticed that his family members were dropping dead at regular intervals. For unknown reasons, Yama, the Lord of Death, was visiting the line every hour to claim one or more family members. On some of Yama's visits one family member dropped dead, and on other visits more than one person dropped dead at once. In the latter case, Yama, being nutty with numbers, ensured that the number of people remaining in the line after any visit was always a non-trivial factor of the number of people that were in the line before that visit. Although Yudhishthira was detached from death and was unafraid, he wondered what is the smallest number of hours he could travel before his entire family, including himself, succumbed to Yama's noose.

*Note: The trivial factors of number N are 1 and N.

Input:
The first line contains T, the number of test cases. The description of T test cases follow. Each test is described by a single integer N, the number of people traveling, in a separate line.

Output:
For each test case, output a single line containing the smallest number of hours before Yama claims all N people.

Constraints:
T <= 100000
1 <= N <= 1000000000

Sample Input
3
1
2
5

Sample Output
1
2
4

Time Limit: 2 Sec

Memory Limit: 256MB

Problem I – Death Row

Problem J – The Sacrificial Firepits

The fire sacrifice at the end of the Mahabharata war required the drawing of two large equilateral triangles on the ground representing Yantras, in which two huge firepits were built up. The royal priests determined the size of the triangles, but their orientation and positions were freely changeable, although they could not overlap (their edges can touch each other though).

Since rain was predicted on the day of the sacrifice, Yudhishthira, the new king, ordered a hastily constructed shelter to be built along with the firepits To save time, the shelter had to be rectangular in design and of minimal area needed to cover both triangles completely.

Tell Yudhishthira, using your considerable skill in Vedic mathematics, what the minimum area of the rectangle should be.

Input:

The first line contains the number of test cases T. Each of the next T lines contains two integers S1 and S2, denoting the sides of the two equilateral triangles.

Output:

For each test case, output the answer rounded to 3 decimal places.

Constraints:

T <= 1000
1 <= S1, S2 <= 100

Sample Input:

2
3 5
10 10

Sample Output:

23.816
129.904

Time Limit: 2 sec
Memory Limit: 256 MB

Problem K – Planetary Alignment

The Kauravas and the Pandavas are playing a game of dice for big stakes, such as their kingdom. Yes, Las Vegas has nothing on them! Since the game is based so much on luck, the Kauravas turn to astrology to predict which is the luckiest time to play. Hence they come to you, the astrology expert.

You observe a far off solar system that has three orbiting planets, with all three orbits coplanar. You know that their time periods are T1, T2 and T3 (where Ti is the time taken by the ith planet to complete one rotation around the sun). Also, you know that at time t = 0, all three were aligned in the same straight line, along with the sun(of that solar system). It is now time t = t0, and you wonder when they would be in a straight line again.

Your findings however show that by the time this happens next, everyone would be dead, and you report this to the Kauravas. They then decide to settle for an approximate alignment as a sign of luck, and shower you with gold to do something, anything, to give them an edge, or else, they say, heads will roll!

The approximate alignment of the planets is one where all the planets are within a sector of 10 degrees, centered at the sun. That is, if the three planets are at angles a1, a2, a3, then max(a1, a2, a3) - min(a1, a2, a3) <= 10 degrees. (Of course, 359 degrees and 9 degrees are also within a 10-degree sector.) You want to find the nearest point of time in the present or future when they will be approximately aligned.

Input:
The first line contains T, the number of test cases. The description of T test cases follow. Each test case is described by one line containing four integers, t0, T1, T2 and T3.

Output:
For each test case print a single line containing the approximate time when the planets approximately align. Relative and absolute errors of 10^-6 are acceptable.

Constraints:
1 <= T1, T2, T3 <= 10000
0 <= t0 <= T1*T2*T3
T <= 2000

Sample Input:

```
3
1 1 1 3
8 5 7 12
1 9 4 5
```

Sample Output:

```
0.458333333
9.013888889
178.800000000
```

Time Limit: 2 sec
Memory Limit: 256 MB