SSL WIRELESS

Problem A

Almost Pattern Matching II

Bangladesh Association of Problem Setters

Two strings are defined to be *almost equal* if they are of the same length and their hamming distance is less than or equal to **K.** What this means is that they can have at most **K** mismatches.

Given a text string **S**, a pattern string **P**, and an integer **K**, count how many substrings of **S** are *almost equal* to the pattern **P** with hamming distance at most **K.**

Input

The first line of the input contains an integer **T**, denoting the number of test cases. The next **T** lines will contain the test cases. For each test case, **S**, **P** and **K** will be given in a single line, separated by a single space and in that order.

Constraints

- **1 ≤ T ≤ 10**
- **1 ≤ |S|, |P| ≤ 1200000**
- **0 ≤ K ≤ 4**

S and **P** will consist of lowercase english letters **(a-z)** only. The sum of all the characters (number of characters in text string and the number of characters in the pattern string across all the test cases) in a single input file will not exceed **2500000.**

Output

For each test case, print the case number followed by the count of substrings in **S** which are almost equal to the pattern **P** with hamming distance at most **K**. Refer to the sample input/output for more clarity on the format.

Sample Input	Output for Sample Input
5 abbabaabbaababbabaababbaa abbab 0 abbabaabbaababbabaababbaa abbab 1 abbabaabbaababbabaababbaa abbab 2 abbabaabbaababbabaababbaa abbab 3 abbabaabbaababbabaababbaa abbab 4	Case 1: 2 Case 2: 8 Case 3: 11 Case 4: 14 Case 5: 19

SSL WIRELESS

Problem B

The Social Network

Bangladesh Association of Problem Setters

Alice wants to build a new social networking site. And you are hired to implement the functionalities of the site.

There are **N** users in the site. Every user can upload posts in the site and he/she can see his/her posts by default. But if a user wants to see someone else's post then both of the user must be in the same network. Two users will be in the same network if they are friends or friends' friends or friends' friends' friends and so on. Once two user **U** and **V** are in the same network then user **U** can see all the posts that user **V** had previously uploaded and going to upload later and vice versa.

You are given **Q** queries. You need to perform the queries in the given order. Each query can be one of these types.

- **0 U V**, means user **U** and user **V** are now friends.
- **1 U T**, means user **U** uploads a post on the site in time **T**. Different users can upload posts at the same time. Also a user can upload multiple posts at the same time.
- **2 U L R**, Count the number of posts user **U** will see from time **L** to **R** inclusive. User **U** will see a post if that post is uploaded between time **L** to **R** inclusive and is uploaded by user **U** or some other user who is in the same network with user **U**.

Input

The first line contains an integer **TC (1 ≤ TC ≤ 4)** denoting the number of test cases. Each test case starts with two space separated integers **N (1 ≤ N ≤ 10^5), Q (1 ≤ Q ≤ 3*10^5)** denoting the number of users and number of queries respectively. Next **Q** lines contains one of these types of queries.

- **0 U V (1 ≤ U, V ≤ N, U ≠ V)**
- **1 U T (1 ≤ U ≤ N), (1 ≤ T ≤ 10^9)**
- **2 U L R (1 ≤ U ≤ N), (1 ≤ L ≤ R ≤ 10^9)**

Output

For each test case, print the case number and for each query of type **2 U L R**, print the number of posts that user **U** will see from time **L** to **R** inclusive in a new line. You can safely assume that the result of a **2 U L R** type query will not be affected by any subsequent queries. Please see sample for details.

Sample Input

```
2
3 6
1 2 7
2 1 1 10
0 2 1
2 1 1 10
1 2 1
2 1 1 10
3 7
1 3 2
1 1 100
1 1 2
0 1 3
0 2 1
2 2 1 10
2 1 2 2
```

Output for Sample Input

```
Case 1:
0
1
2
Case 2:
2
2
```

SSL WIRELESS

Problem C

ICGeSi Standings

Bangladesh Association of Problem Setters

"Programming contest fosters creativity, teamwork, and innovation in solving critical problems, and enables students to test their ability to perform under pressure."

ICGeSi maintains a simplified contest ranklist that has only 3 columns consisting of team id, the number of problems solved by that team and their total penalty time. To make the contest more thrilling, the contest ranklist is frozen for the last 1 hour, no update is shown.

The contest, ranklist and result is determined by the following rules:

- The team with the higher number of solved problems will get a higher rank. Rank 1 is considered as the highest rank.
- In case of a tie between two teams with the same number of problems solved, the higher rank is determined by the penalty time. The team with the lower penalty time will get the higher rank.
- The penalty time for a particular problem is determined by the following formula:
  - If a problem is not solved, then the penalty time for that problem is **0 (zero)**.
  - The time consumed for a solved problem to be solved is considered as its penalty time. For example, if a problem is solved at the 20^{th} minute of the contest, then the penalty time for that problem is, P = 20.
  - Total Penalty for a team is the summation of penalties for all problems.
    $P_{TEAM} = P_A + P_B + P_C + \ldots$, where problems are named as A, B, C, … in alphabetic order.
  - A team can not submit for a particular problem after getting it accepted once.
- After considering the number of solved problems and penalty time, if there is still a tie, teams will share a common rank. For example, let's assume that, both TeamX and TeamY solved 2 problems with total penalty time $P_{TeamX} = P_{TeamY} = 220$. It is a tie. So, they will share a common rank. In the ranklist, TeamX may come before TeamY, or TeamY may come before TeamX. Both are valid. The same also holds when more than 2 teams are tied.
- You may safely assume that each team solved at least one problem before the ranklist was frozen.

It is ICGeSi DRPC-2019 (Dhaka Regional Premier Contest, 2019). This is one of the biggest competitions in the history of DRPC and is a high voltage contest. All the problems are critical and require a lot of analysis and coding skills to be solved. For any team anything is possible, the *agony* and the *ecstasy*.

The contest has already finished and the jury is ready to unfreeze the rank list and publish the result. The jury of the contest posted an interesting statistic: "**in ICGeSi DRPC-2019, there is no such team who got an accepted verdict after any of their submissions had been rejected during the frozen hour.** " For example, let's assume there are 5 submissions from a particular team in the frozen hour. If their 2nd submission was rejected, we know for sure that the following 3 submissions were rejected too.

Everyone is eagerly waiting for the result. The organizer decided to arrange a closing ceremony in order to address the winners, participants, supporters, owners, sponsors and stakeholders. It's been five hours after the contest, but the ceremony is yet to start. A lot of rumours are going on. A lot of criticizations are also being made. From the judges' perspective, there are two types of critics: positive critics, negative critics. Negative critics are claiming that the jury is taking time to actually manipulate the results, while positive

critics are claiming that the organizers are not ready as the guests are yet to arrive. A group of people is also claiming that they have the result which has been leaked from judgeroom.

You are given the frozen ranklist of ICGeSi DRPC-2019. For each team you also know when they submitted solutions during frozen hour but you do not know their verdicts (whether they were accepted or rejected). You are also given the result which is claimed as being leaked from the judges room. You have to determine whether such a final ranklist is possible or not!

Input

Input starts with an integer **K** (**$1 \leq K \leq 750$**) in a single line denoting the number of test cases. Each of the **K** test cases starts with an integer **N** (**$1 \leq N \leq 50$**) denoting the number of teams participating in the contest. Each of the next **N** lines describe teams starting with four integers **Id (**$1 \leq Id \leq N$**)**, **S (**$1 \leq S \leq 120$**)**, **TP (**$1 \leq TP \leq 7260$**)** and **M** (**$0 \leq M \leq 60$**) denoting team id, number of problems solved by them, their total penalty time before the ranklist was frozen and the number of submissions during the frozen hour respectively. Then in the same line, there will be **M** integers T_1, T_2, …, T_M (**$121 \leq T_1 < T_2 < T_3 < ... < T_M \leq 180$**) denoting the time of their submissions in frozen hour. **Summation of M over all test cases $\leq 1.5 \times 10^6$.**
Next line contains **N** space separated integers representing team Id from higher rank to lower rank in the leaked result (from left to right).

Output

For each test case, print the case number followed by a string "**Say no to rumour >:**" if no such result is possible, otherwise print "**We respect our judges :)**" in a single line. Check the sample I/O section for more clarity.

Sample Input

```
2
3
2 3 100 1 130
3 2 110 2 150 160
1 1 110 3 170 175 180
1 2 3
2
1 2 100 0
2 1 110 1 130
2 1
```

Output for Sample Input

```
Case 1: We respect our judges :)
Case 2: Say no to rumour >:
```

Explanation:

Case 1: A possible solution is as follows.
Team Id = 1 (4/635) , Team Id = 2 (3/100), Team Id = 3 (2/110).

Case 2: There is no way team Id = 2 can get a higher rank than team Id = 1. So, it's a rumour.

Problem D

ICPC Standings

Competitive programming is a sport that requires the elegance and the art of problem solving. ICPC, the premier global competition for programming, aims to recognize the best and the brightest young programmers from universities around the world.

Duration of an ICPC style contest is five hours. To make the contest more thrilling, the contest ranklist is frozen for the last 1 hour; that is accepted/rejected verdict is not shown in the rank list, but we can see how many times a problem is submitted by a particular team. After the contest the jury publishes the result and acknowledges the winners.

In ICPC style contest, ranklist and result is determined by the following rules:

- The team with the higher number of solved problems will get a higher rank. Rank 1 is considered as the highest rank.
- In case of a tie between two teams with the same number of problems solved, the higher rank is determined by penalty time. Team with lower penalty time will get the higher rank.
- Penalty time for a particular problem is determined by the following formula:
    - If a problem is not solved, then penalty time for that problem is **0 (zero)**.
    - Penalty for a solved problem, $\mathbf{P_A = (NS - 1) \times 20 + ST}$
      Where, **NS = Number for Submission for that problem, ST = Last Submission Time**
    - Total Penalty for a team is the summation of penalties for all problems.
      $P_{TEAM} = P_A + P_B + P_C + \dots$, where problems are named as A, B, C, … maintaining alphabetic order.
    - A team can not submit for a particular problem after getting it accepted once.
- After considering the number of solved problems and penalty time, if there is still a tie, teams will share a common rank. For example, let's assume that, both Team X and Team Y solved 2 problems with total penalty time $P_{Team\ X} = P_{Team\ Y} = 330$. It is a tie. So, they will share a common rank. In the ranklist, Team X may come before Team Y, or Team Y may come before Team X. Both are valid. The same also holds when more than 2 teams are tied.

A frozen rank list consists of **M+2** columns where **M** denotes the number of problems.
First Column: Contains Team Name and fixed column having width of twenty characters.
Second Column: Contains Number of Solve and Penalty time and fixed column having width of seven characters. It can be expressed as a fixed format **c/d**, where c is the total number of solve ($\mathbf{c \geq 1}$) and **d** is the total penalty time.
Third Column to (M+2)th column: may have any of the following formats and has fixed column width of seven characters.

- **0/-- :** This problem is not attempted.
- **c/d :** $\mathbf{1 \leq c \leq 20}$ and $\mathbf{1 \leq d \leq 240}$ means that the problem was solved in the c^{th} attempt at the d^{th} minute.
- **-c/-- :** $\mathbf{c<0}$ and $\mathbf{|c| \leq 20}$, means that the problem was submitted c times and all the submissions were rejected.
- **?c/d :** $\mathbf{1 \leq c \leq 20, 241 \leq d \leq 300}$, means that the problem was last submitted at the d^{th} minute during contest and it was their c^{th} submission on that problem. Its verdict is unknown as it was submitted in the frozen hour.

Verdicts are instantaneous and during frozen hours teams are notified only about their own submission verdict (accepted/rejected).

Team Name	Total	A	B	C	D
Team X	2/330	?10/255	0/--	1/10	5/240
Team Y	2/330	5/230	0/--	1/20	?5/270
Team Z	1/30	-4/--	1/30	?1/241	?1/245

The above table is an example of a frozen rank list, Team X had solved two problems before the rank list was frozen. They solved Problem C with penalty time P_C = (1-1)×20 + 10 =10 and Problem D with penalty time P_D = (5-1)×20 + 240 = 320. So, total penalty for Team X, $P_{Team\ X}$ = 10+320 = 330. They also submitted for problem A at 255 minutes and it was their 10^{th} submission on that problem. We don't know the verdict of 10^{th} submission as the rank list is frozen. They did not attempt problem B.

It is ICPC Dhaka Regional Preliminary Contest, 2019. The contest is already finished and the jury is ready to unfreeze the rank list and publish the result. Everything was going well. But, the judge server crashed and all of a sudden things got messy. They have lost information about which submissions were accepted and which submissions were rejected. But they have the final result consisting of one column (team names only) from higher rank to lower rank (Only the first column of the final unfrozen ranklist) .
In this problem, you have the frozen rank list and the judges' final result. You have to figure out how many different ways this final result can be achieved after unfreezing the rank list.

For example let the following be the final result of the above example contest:

```
Team Z
Team X
Team Y
```

It can be achieved in the following ways,

Team Z	2/271	-4/--	1/30	1/241	-1/--
Team X	2/330	-10/--	0/--	1/10	5/240
Team Y	2/330	5/230	0/--	1/20	-5/--

Team Z	2/275	-4/--	1/30	-1/	1/245
Team X	2/330	-10/--	0/--	1/10	5/240
Team Y	2/330	5/230	0/--	1/20	-5/--

Team Z	3/516	-4/--	1/30	1/241	1/245
Team X	2/330	-10/--	0/--	1/10	5/240
Team Y	2/330	5/230	0/--	1/20	-5/--

```
|--------------------|-------|-------|-------|-------|-------|
|Team Z              |3/516  |-4/--  |1/30   |1/241  |1/245  |
|Team X              |3/765  |10/435 |0/--   |1/10   |5/240  |
|Team Y              |2/330  |5/230  |0/--   |1/20   |-5/--  |
|--------------------|-------|-------|-------|-------|-------|
```

Input

Input begins with an integer **T** in a single line denoting the number of test cases. Each of the **T** test cases starts with two integers **N** and **M** in a single line denoting the number of teams and the number of problems respectively. Then there will be **(N+1)** $\times$ **2 + 1** lines of formatted input displaying the tabular format of the rank list. Among these

- Each odd numbered line will display the table border.
- Each even numbered line will contain M+2 columns. All these columns will be separated by '|' and they have their fixed width as mentioned in the statement above.

Then there will be a blank line.
Next N+2 lines represent the judges' final result, a single column containing team names (with the fixed width of twenty characters) from higher rank to lower rank in tabular format.

You may safely assume that:

- Table alignment will be exactly the same as it is in the sample input.
- All the values in the table will be left-aligned.
- No two teams have the same name.
- The team name consists of lowercase and uppercase Latin letters (a-z), ('A'-'Z') and white space(s) only.
- The name consists of less than 21 characters.
- At least one problem is solved by all the teams before the ranklist was frozen.

Constraints

1 $\leq$ **T** $\leq$ **50**
1 $\leq$ **N** $\leq$ **100**
1 $\leq$ **M** $\leq$ **12**

No problem is submitted more than 20 times by a particular team.

Output

For each test case, print the case number followed by the number of ways the final result can be achieved. Since the answer can be large, print the answer modulo **1000000007** (**10^9+7**).

Two ranklists RANKLIST1 and RANKLIST2 are considered different if for at least one team any of the (M+1) columns from 2nd Column to M+2th column has different values in the two ranklists.

Sample Input

```
1
3 4
|--------------------|-------|-------|-------|-------|-------|
|Team Name           |Total  |A      |B      |C      |D      |
|--------------------|-------|-------|-------|-------|-------|
|Team X              |2/330  |?10/255|0/--   |1/10   |5/240  |
|--------------------|-------|-------|-------|-------|-------|
|Team Y              |2/330  |5/230  |0/--   |1/20   |?5/270 |
|--------------------|-------|-------|-------|-------|-------|
|Team Z              |1/30   |-4/--  |1/30   |?1/241 |?1/245 |
|--------------------|-------|-------|-------|-------|-------|

|--------------------|
|Team Z              |
|Team X              |
|Team Y              |
|--------------------|
```

Output for Sample Input

```
Case 1: 4
```

Problem E

Palindrome Again? Arghh!

One day Leo was teaching Palindrome to his younger brother Neo. A palindrome is a word that reads the same backward as forward. He also showed him some palindromic words such as "**ototo**", "**madam**" and "**racecar**". Now he gave Neo the following simple problem to check whether he was inattentive during his lesson.

Neo will be given a string **s** of length **N**. The string **s** consists of only the first **K** lowercase English letters. He is allowed to make some changes to the given string. In one single operation, he can choose any two positions **u** and **v (1 ≤ u, v ≤ N)** and swap the characters at those two positions. He has to find the minimum number of operations needed to make the string palindrome.

Input

The first line contains a single integer **T (1 ≤ T ≤ 200),** which denotes the number of test cases. The first line of each test case contains two integers **N (1 ≤ N ≤ 50000)** and **K (1 ≤ K ≤ 5)**, and the second line of each test case contains the string **s**.

Output

For each test case, print a single line in the format "**Case X: R**" without the quotes where **X** denotes the case number and **R** denotes the minimum number of operations Neo needs to apply to make the string palindrome. If there's no solution then **R = -1**.

Sample Input	Output for Sample Input
5 7 4 abcdcab 8 2 abababab 5 3 abcba 6 4 abbccd 9 5 aacdedbbc	Case 1: 1 Case 2: 2 Case 3: 0 Case 4: -1 Case 5: 2

Problem F

Elevators

There are **n** elevators in a row. The elevators are numbered **1** to **n** from left to right. Each elevator has a maximum height **h[i],** which indicates the maximum height it can go upwards. Each elevator starts at time **0** from height **0**. Then they go upward at a speed **1** unit/second. After reaching height **h[i]**, they go downward at a speed **1** unit/second till they reach height **0**. Then they go upward again and the process continues. The elevators never stop after they are started.

Alice is standing on elevator **1** at time **0**. She wants to go to elevator **n**. She can jump from elevator **i** to elevator **i+1** if the current height of elevator **i** is **greater than or equal to** current height of elevator **i+1**, **and** the time is an integer number of seconds. It takes Alice **1** second to make a jump. Notice that, the height condition has to be met when Alice initiates the jump, not when the jump has been made. She can decide to wait for any number of seconds she wants before going for the jump.

Now Alice asks you when is the earliest possible time she can reach elevator **n**.

Input

The first line contains a single integer **T (1 ≤ T ≤ 5)** indicating the number of test cases. Each test case will have a single integer **n (2 ≤ n ≤ 10^5)**, denoting the number of elevators. Next line contains **n** space separated integers, **i-th** of which is the value of **h[i] (1 ≤ h[i] ≤ 10^9)** as described in the statement.

Output

For each test case, print the case number and the answer to Alice's question in a single line. Please see sample for details.

Sample Input

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

Output for Sample Input

```
Case 1: 2
Case 2: 8
```

Explanation of the First Sample

Height of the elevators at second **0**, **1**, and **2** are as below:

Elevator **1** - **0**, **1**, **2**
Elevator **2** - **0**, **1**, **0**
Elevator **3** - **0**, **1**, **2**

At second **0**, Alice is standing on elevator **1**. Elevator **1** is at height **0** and elevator **2** is at height **0**. So the conditions for jump is met and she makes the jump to elevator **2**. She reaches elevator **2** at second **1**. By similar logic, she can make jump from elevator **1** to elevator **2** at second **1**. She reaches elevator **2** at second **2**. This is the earliest time she can reach elevator **2**.

SSL WIRELESS

Problem G

Pairs Forming GCD

Bangladesh Association of Problem Setters

Given **N** and **P**, find maximum **G**, such that there exists at least **P** pairs of integers **(X, Y)**, where **1 ≤ X ≤ Y ≤ N** and **GCD(X, Y)** is equal to **G**.

Here **GCD(X, Y)** indicates the **Greatest Common Divisor (GCD)** of **X** and **Y** - the largest positive integer that divides both **X** and **Y**.

Input

The first line of input contains an integer **T (1 ≤ T ≤ 10^5)**, indicating the number of test cases. The following **T** lines contains two integers **N (1 ≤ N ≤ 10^7),** and **P (1 ≤ P ≤ 10^{15})** each.

Output

For each test case, print the test case number, starting from 1, and an integer, indicating the maximum possible **G**, under the given description. If there is no possible **G**, for given **N** and **P**, then print **-1** instead.

Sample Input	Output for Sample Input
4 25 12 48 125 12 60 16661 5700	Case 1: 4 Case 2: 2 Case 3: -1 Case 4: 121

Explanation of Sample I/O

In case-1, there are 12 possible pairs with G = 4.
In case-2, there are 180 possible pairs with G = 2.
In case-3, there are no possible G for which there can exist at least 60 different pairs.
In case-4, there are 5770 possible pairs with G = 121.

SSL WIRELESS

Problem H

Have You Heard of Cricket?

Bangladesh Association of Problem Setters

Have you heard of Cricket? Not the insect, we are talking about the sports! Don't worry if you haven't heard of it, because we will talk about all the rules of cricket that you need to know for this problem. Even if you know all the rules, please go through the rules again, because some of the rules may have been modified from their actual versions, for this problem.

- Two teams play against each other in a match of Cricket.
- There are two innings. In the 1st innings, one team bats, another team bowls. Then in the 2nd innings, the bowling team of the 1st innings bats, and the other team bowls.
- At first, a toss is held to decide which team will bat first.
- In each innings, 300 valid balls need to be bowled. There is an exception described in the latter points where it is not necessary to bowl 300 valid balls in an innings.
- The batting team has 10 wickets.
- Any integer value from **0 to 7** can be scored in a single ball. Also the batting team can lose a wicket in a ball. If a wicket is lost, 0 run is scored from that ball.
- Sometimes, the bowler may bowl a no-ball. A no-ball is an invalid ball. One run is added to the score of the batting team for a no-ball. The bowler has to bowl the ball again. For example, if the first ball of the innings is a no-ball and the batsman scores 4 runs from the ball, then the batting team will have 5 runs from 0 ball.
- If the batting team loses all the 10 wickets, then their innings will immediately end, even if 300 valid balls have not been bowled.
- The team batting in the 2nd innings has to score at least one run more than the score of the 1st innings to win. If they score the same as that of the 1st innings, then the match will be tied. If they score less than that of the 1st innings, then they will lose.
- The first innings will end when 300 valid balls have been bowled or the batting team loses 10 wickets.
- The second innings will end if 300 valid balls have been bowled or the batting team loses 10 wickets or their score becomes more than that of the first innings.

Team **X** and **Y** are playing in a cricket tournament. There is only one more match left in the group stage which will be held between **X** and **Y**. Only one team between **X** and **Y** will qualify to the next round. If **Y** wins the match or the match is tied, then **Y** will qualify to the next round. But even if **X** wins the match, their point will become equal to that of **Y**. So if **X** wins the match, **net run rate (NRR)** will decide which team will go to the next round. The team having the larger **NRR** will qualify to the next round. If **NRR** of both the teams is equal, then **Y** will qualify because of their current ranking. **NRR** of a team is calculated in the following way:

$$\textbf{NRR of a team} = \frac{Total\ runs\ for}{Total\ balls\ for} - \frac{Total\ runs\ against}{Total\ balls\ against}$$

For example, let's assume that team T has scored 504 runs and played 900 balls in all of its matches. In those matches, its opposition teams scored 759 runs and played 700 balls in total. Then team T's NRR will be calculated like this:

$$\textbf{NRR of team T} = \frac{504}{900} - \frac{759}{700} = \textbf{-0.5243}$$

If the batting team loses all their wickets before the completion of 300 balls, then the value of **"balls for"** of the batting team and **"balls against"** for the bowling team for that match will be **300**. If the team batting second wins the match in less than 300 balls, then the value of **"balls for"** of the batting team and **"balls against"** for the bowling team for that match will be exactly the number of balls the team batting second played.

You will be given all the data that you need to calculate the NRR of team **X** and **Y** before the last match, the result of the toss i.e. which team will bat first in the last match. You will have to find out the lowest number of runs the team batting in the **1st** innings can score after which, **X** can theoretically still qualify.

Input

In the first line, the number of test cases **T** will be given. Then for each case, three lines will be given. The first line will contain four integers signifying "**total runs for"**, "**total balls for"**, "**total runs against"** and "**total balls against"** of team **X** respectively. The second line will contain the same information for team **Y** in the same order. You can calculate the NRR of both the teams using these data. Then, the last line will contain a character **C** signifying which team will bat first.

Constraints

$1 \le T \le 3*10^5$
1 ≤ any integer in input (except for T) ≤ 10^4
C belongs to {'X', 'Y'}

Output

For each case, print the case number and then the least number of runs that the team batting first can score after which **X** can still theoretically qualify. If there is no way for **X** to qualify, print "**-1**" instead. Please see the sample for details.

Sample Input

Output for Sample Input

Sample Input	Output for Sample Input
2 504 900 759 700 10 100 1000 200 X 504 900 759 700 10 100 1000 200 Y	Case 1: 1 Case 2: 0

Note:

For the first test case of the sample input, X is batting first. Let's assume X scores 1 run in the 1st innings. Then theoretically it is possible that Y loses all 10 wickets for 0 run in the 2nd innings. If that happens, then X will win the match and they will have a better NRR than that of Y after the match. Considering the above scenario takes place in this match, the calculation of NRR of X and Y after the match are shown below:

NRR of X = $\frac{504+1}{900+300} - \frac{759+0}{700+300}$ **= -0.338**

NRR of Y = $\frac{10+0}{100+300} - \frac{1000+1}{200+300}$ **= -1.977**

Since, NRR of X is better than Y, X will qualify to the next round.

If X had scored less than 1 run, i.e. 0 run, then the match would have been either tied or won by team Y.

Problem I

Almost Forgot to Welcome

ICPC Dhaka Regional is the biggest programming competition in Bangladesh. Also the most anticipated one as well. Students from all the different universities storm their brains all year in preparation for this competition. You are now taking part in this competition, so a big congratulations to you.

Dhaka site is one of the biggest sites in ICPC. This year almost 1800 teams are participating in the competition. So in celebration, we are going to give you an easy problem to solve.

You have to write a program, which will print the line “Welcome to ICPC Dhaka Regional Online Preliminary Contest, 2019” (without quotes).

Note: you can’t output anything other than the required output, and the line must end with a newline (‘\n’). Take special care about spelling and case. If you alter any of those, you may not get accepted.

For your convenience, we are providing one sample program in C/C++ which prints “Bangladesh”. You just have to change the code to your requirement.

```
#include <stdio.h>
int main()
{
    printf("Bangladesh\n");
    return 0;
}
```

SSL WIRELESS

Problem J

Good Things Come to Those Who Wait

Bangladesh Association of Problem Setters

Finding divisors of an integer is common task. We all did it in some time or other during our programming courses. We know that there are several ways to find divisors of an integer **N**. All of them involve some kind of programming using loops. But what if we ask you to solve the reverse problem?

Given all the proper (all the divisors except **1** and **N**) positive divisors of a positive composite integer **N**, you need to find the value of **N**.

Input

The first line of the input contains an integer **T (0 < T < 11)**, denoting the number of test cases. The next **T** lines will contain the test cases. For each test case an integer **K (K > 0)** will be given which is the number of proper positive divisors of **N**. The following line will consist of **K** integers. These are all the proper positive divisors of **N**. You should assume that the divisors will be given in such a way so that the value of **N** will always be greater than 1 and less than or equal to 10^{12}.

Output

For each test case, print a single line in the format **"Case X: N"** (without the quotes) where X denotes the case number and **N** is the answer. Refer to the sample input/output for more clarity on the format.

Sample Input	Output for Sample Input
3 2 2 4 1 3 2 5 2	Case 1: 8 Case 2: 9 Case 3: 10