Tech Fest 2014

11th IIUC Inter-University Programming Contest

I I U P C 2 0 1 4

Problemset

Allah grant me the serenity
To accept the problems that I cannot solve
The persistence to solve the problems that I can
And the wisdom to know the difference

Organized by: Faculty of Science and Engineering
International Islamic University Chittagong

IIUPC 2014

Problem A: The Soldier's Dilemma

N soldiers are preparing for their parade. Each of them has different heights and they are standing in a line. Their commander will order any three of them to step forward. All the other soldiers will then leave the ground and selected three will stay. Now if from left to right these three soldier's order is ***short, tall, medium*** then their commander will become furious and punish them.(Even if they are "medium, tall, short" or in any other order from left to right they will not be punished) Now the soldiers don't want to be punished. In how many ways can the **N** soldiers stand in a line that no matter which three are chosen there order will never be ***short, tall , medium*** from left to right?

Input

The first line contains number of test case **T** (**$1 \leq T \leq 500$**). Each of the next **T** lines contains an integer **N** (**$1 \leq N \leq 5000$**).

Output

For each of the test case you must output the number of ways **N** soldiers can stand in a line. As the number can be very large output it modulo **1000000007** (**10^9+7**).

Sample Input	Output for Sample Input
2 **2** **3**	**2** **5**

Problem Setter: Sakib Shafayat

IIUPC 2014

Problem B: Count LCM

LCM is an abbreviation used for **L**east **C**ommon **M**ultiple in Mathematics. We say **LCM (a, b) = L** if and only if **L** is the least integer which is divisible by both **a** and **b**.

You will be given **N, M**. You have to count number of pair (**i**, **j**) such that **LCM (i, j) = i × j**, where **1≤i≤N** and **1≤j≤M**.

Input

Input starts with an integer **T (≤ 1000)**, denoting the number of test cases.

Each case starts with a line containing two integers **N, M (1 ≤ N, M ≤ 10^9,** and **minimum of (N, M)** ≤ **10^6**).

Output

For each case, print number of such pair.

Sample Input	Output for Sample Input
3 1 2 4 2 3 5	2 6 12

Problem Setter: Mohammad Hafiz Uddin
Alternate Solution: F. A. Rezaur Rahman Chowdhury

IIUPC 2014

Problem C: Happy Birthday Tutu

Tutu is a jolly little kitten who is the biggest fan of stone games. His elder brother Milo designed a game for him on the occasion of his 2nd birthday. But Tutu needs another player to play with him, otherwise he will get sad, even on his birthday! ☹ You, being a good old fella, can't let that happen. So you decided to play with Tutu. But none of you want to lose, so you both play optimally, but as it is his birthday, you let Tutu make the first move.

The game consists of **K** piles, each pile has $\mathbf{A_i}$ **(1≤i≤K)** stones. In each turn, a player can select any number of piles but not all piles (i.e. at most **K-1** arbitrary piles) and take any number of stones from all of the selected piles (you can take stones from multiple piles in single move). Note that, the player cannot make an empty move, i.e. at least one stone must be taken in each move. The player, who cannot make a move, loses the game.

Given the number of piles and the number of stones in each of the piles, can you determine who wins the game, given that both of you play optimally?

Input

First line of input file contains number of test cases, **T ≤ 50** and **T** cases follow. Each case starts with an integer **K (1≤ K≤ 100000)**, number of piles. Next line contains **K** integers, $\mathbf{A_1, A_2, \ldots A_K}$ **(0≤A_i≤ 100).**

Output

Output consists of T lines, one for each case. For each case, if Tutu wins, output "**Happy Birthday Tutu!**", otherwise print "**Better luck next time!**" (Quotes for clarity).

Sample Input	Output for Sample Input
3 1 5 6 1 2 3 4 5 6 4 0 0 0 0	Better luck next time! Happy Birthday Tutu! Better luck next time!

Problem Setter: Sadia Nahreen

IIUPC 2014

Problem D: Easy Peasy

Given an array of integers, find the number of the segments of the array without repeating integers in the segment. For example, number of the segments of the array without repeating letters for **{1,2,1}** is **5**. In 0 based indexing, these segments are **(0,1), (1,2), (0), (1), (2).**

Input

Input starts with **T (≤ 20)**, the number of test cases to follow.
For each test case, in the first line one integer **N(1≤N≤100000)** is given. **N** is the number of integers in the array. In the next line **N** integers separated by space is given. **i^{th}** integer will be **arr[i] (0≤arr[i] ≤1000000000).**

Output

For each case, print the number of the segments of the array without repeating integers in the segment.

Sample Input	Output for Sample Input
2 3 1 2 1 5 1 2 3 1 2	5 12

Problem Setter: F. A. Rezaur Rahman Chowdhury
Alternate Solution: Mohammad Hafiz Uddin

IIUPC 2014

Problem E: Risk of Trading

Once upon a time, there were two great countries separated by a great river. They were called East country and West country. There were many great cities in both countries. And trading between them was profitable. Everybody lived in peace, but everything changed when the fire pirates attacked. They intercepted the ships across the river and looted them.

Then a big trading company got in trouble. There were **N** cities in both of the countries. And there was a ship in each of the **N** cities of West countries. They wanted to send each of the **N** ships to one of the different cities of the East country. Thus, after the voyage, each of the **N** cities in the East country would have one of the ships. But the pirates were in the river. And they could attack any ship. Even if one of the ships got attacked, the company would have faced great problems. Now from the reports of pirates attack, they had calculated the risks of sending a ship from each of the cities of West country to each of the cities of East country. They just needed to find a safe pairing between each **N** cities of West country to one of the **N** city of East country such that the risk of even one of the ships get attacked would be minimum. Could you help them?

Input

First line of the input is **T (T≤20),** which is the number of test cases in input. Then **T** test cases follow. In each of the test case the first line of the input is **N (1≤N≤40)** the number of cities in each countries. Then next **N** lines are for each of the cities of West country. In $\mathbf{i}^{th}$ line, there are **N** probabilities; $\mathbf{j}^{th}$ of this is the probability of getting attacked by pirates if they sail from $\mathbf{i}^{th}$ city of West country to the $\mathbf{j}^{th}$ city of East country. The probabilities are given in fraction form in **a/b** format where **0≤a≤b≤40** and **a/b** is in irreducible format (i.e. a and b is co-prime).

Output

For each of the cases, print the minimum probability of one the ship getting attacked by the pirates. Final answer's a and b will fit in a signed 64 bit integer number (long long in C/C++, long in Java).

Sample Input	Output for the Sample Input
2 2 1/2 1/2 1/4 3/4 2 1/8 7/8 0/1 1/15	5/8 22/120

Problem Setter: Ridowan Muhammad
Alternate Solution: Bidhan Roy

Output Explanation

In the both cases there can be two type of pairing,
- City 1 of West country to city 1 of East and city 2 of West country to city 2 of East.
- City 1 of West country to city 2 of East and city 2 of West country to city 1 of East.

In first pairing,
- The probability of the ship from city 1 of West getting attack but not the ship from city 2 is 1/2 * (1 – 3/4) = 1/2 * 1/4 = 1/8.
- The probability of the ship from city 2 of West getting attack but not the ship from city 1 is (1 – 1/2) * 3/4 = 1/2 * 3/4 = 3/8.
- The probability of both of the ship getting attacked is 1/2 * 3/4 = 3/8.

So total risk of one of the ships getting attacked is 1/8 + 3/8 + 3/8 = 7/8.

In second pairing the risk is 1/2 * (1 – 1/4) + (1 – 1/2) * 1/4 + 1/2 * 1/4 = 3/8 + 1/8 + 1/8 = 5/8.

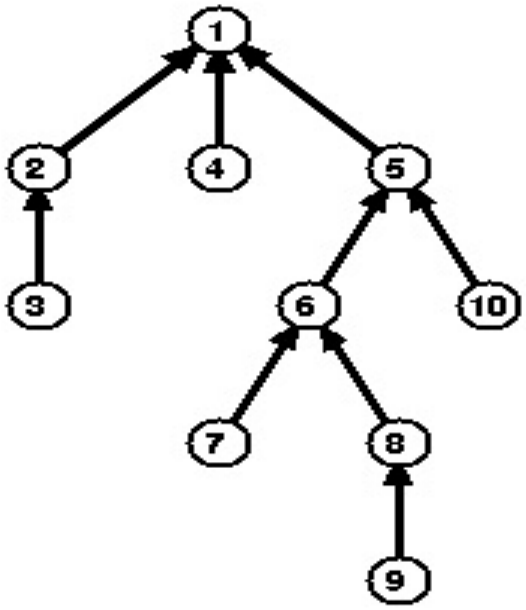
So the second pairing gives minimum risk 5/8.

In the second example, for the first pairing risk is 1/8 * (1 – 1/15) + (1 – 1/8) * 1/15 + 1/8 * 1/5 = 14/120 + 7/120 + 1/120 = 22/120 = 11/60.
For the second pairing risk is 7/8 as the ship from city 2 of west to city 1 of East will go risk free(0/1 = 0 risk). But 11/60<7/8 so the answer is 11/60.

IIUPC 2014

Problem F: Light Combat Aircraft

In graph theory. the **lowest common ancestor** (**LCA**) of two distinct nodes ***v*** and ***w*** in a rooted tree is the lowest (i.e. deepest) node that has both ***v*** and ***w*** as descendants, where we define each node to be a descendant of itself (so if ***v*** has a direct connection from ***w***, ***w*** is the lowest common ancestor).



For example, on the above tree (depicted from case 1) **LCA(3,5) = 1**, **LCA(7,10) = 5**, **LCA(6,5) = 5** etc.

In this problem, given a Forest, i.e. a disjoint union of rooted trees, you have to find out for each node **u** how many distinct pair of nodes **(*v*,*w*)** exist such that **LCA(v,w)** would be **u**. You should assume that both **(*v*,*w*)** and **(*w*,*v*)** are same pair.

Input

First line of input file contains number of test cases, **T ≤ 100** and **T** cases follow. Each case starts with an integer **N (1≤ N≤ 10000)**, number of nodes in the forest. Next line contains **N** integers, $p_1, p_2, \ldots p_N$ **(0≤ p_i≤ N),** where p_i is the parent of i^{th} **(1≤ i≤ N)** node in a rooted tree of the forest , If $p_i = 0$ then node **i** is a root in rooted tree.

Output

For each case, print the forest number starting from **1** and number of LCA pair for each node (ordered by node number) separated by space. See the sample output for exact formatting.

Sample Input	Output for Sample Input
4 10 0 1 2 1 1 5 6 6 8 5 3 2 0 0 4 0 1 2 1 4 0 1 0 3	Forest#1: 29 1 0 0 9 5 0 1 0 0 Forest#2: 0 1 0 Forest#3: 5 1 0 0 Forest#4: 1 0 1 0

Problem Setter: Prasanjit Barua
Alternate Solution: Kayser Abdullah

Output Explanation

In **case 2**, in the given forest among the two trees rooted at **2** and **3**, there is no pair for which LCA is **1** or **3**. For pair **(1, 2)** LCA is **2**. So, total pair for **2** is **1**.
In **case 3**, for pair **(1,2), (1,3), (1,4), (2,4), (3,4)** LCA is **1**. For only pair **(2,3)** LCA is **2**. There is no pair for which LCA is **3** or **4**.

Note: Dataset is huge, so use faster I/O methods.

I I U P C 2 0 14

Problem G: Count It

Following is a code in C.

```
#include <stdio.h>
int num[1000006];

int main()
{
     int i,n,cas;

     num[0]=0;
     for(i=1;i<=1000000;i++) num[i]=num[i/2]+(i%2);

     scanf("%d",&cas);

     while(cas--)
     {
          scanf("%d",&n);
            printf("%d\n",num[n]);
     }

      return 0;
}
```

This code will work fine for values of n up to 10^6. But for higher value of ***n***, the code will not work for memory, time constraints. You have to write a code which will give identical result for higher values of ***n.***

Input

The first line contains number of test case ***T*** (**1≤T≤500**). Each of the next ***T*** lines contains an integer ***n*** (**1≤*n*≤10^{18}**).

Output

For each of the test case you must output the answer in a line.

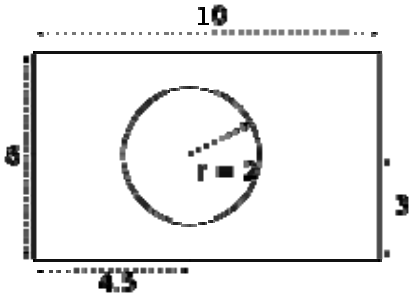
Sample Input	Output for Sample Input
3 4 5 6	1 2 2

Problem Setter: Sakib Shafayat

IIUPC 2014

Problem H: Perfect Flag

The national flag of Bangladesh is bottle green in color and rectangular in size with the **length (L)** to **width** ratio of **10:6**. It bears a red circle on the background of green. It maintains the **length (L)** to **radius** ratio of **5:1** (If the **length** is **10** then **width** should be **6** and **radius** should be **2**). The color in the background represents the greenery of Bangladesh while the red circle symbolizes the rising sun and the sacrifice of lives in our freedom fight.
Its centre will be placed on the intersecting point of the perpendicular drawn from the **nine-twentieth** part of the length of the flag, and the horizontal line drawn through the **middle** of its width.



Input

First line of input will contain the number of test cases, **T ≤ 100.** Then there follows **T** lines, each containing seven positive integer **x0**, **y0**, **x1**, **y1**, **cx**, **cy**, **r ≤ 1000**. Here,
(x0,y0) is the lower left corner of the rectangle
(x1,y1) is the upper right corner of the rectangle
(cx,cy) is the center of the circle and **r** is the radius.

Output

For each test case if the given data represents a valid flag design print 'YES' else print 'NO'.

Sample Input	Output for Sample Input
4 0 0 20 12 9 6 4 0 0 10 6 4 3 2 1 1 21 13 10 7 4 0 0 20 20 9 10 4	YES NO YES NO

Note: Pi is considered to be **acos(-1).**

Problem Setter: Kaysar Abdullah

IIUPC 2014

Problem I: Armstrong Number

A number **N** is an ***Armstrong number of order*** **n** (**n** being the number of digits) if

abcd … = $a^n + b^n + c^n + d^n + \ldots = N$

For example, 153 is an Armstrong number of order 3 because

$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$.

Likewise, 54748 is an Armstrong number of order 5 because

$5^5 + 4^5 + 7^5 + 4^5 + 8^5 = 3125 + 1024 + 16807 + 1024 + 32768 = 54748$.

In this problem you have to determine whether a given number is Armstrong number or not.

Input

The first line of input is an integer, **T** that determines the number of test cases. Each of the next **T** lines contain a positive integer **N**, where **N ≤ 1000000000**.

Output

For each line of input, there will be one line of output. If **N** is an Armstrong number print "Armstrong", otherwise print "Not Armstrong" (without the quotes).

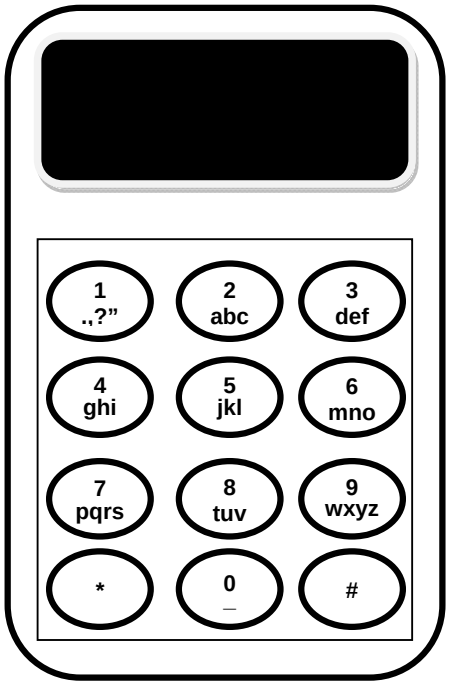
Sample Input	Output for Sample Input
3 153 2732 54748	Armstrong Not Armstrong Armstrong

Problem Setter: Mohammed Shamsul Alam
Alternate Solution: Tanveer Ahsan

IIUPC 2014

Problem J: Mobile SMS

Ali is known as very much stingy. He bought an old second-hand mobile phone. He usually uses the mobile phone to receive phone calls from others and rarely makes a phone call. Suddenly, he observes that sending a mobile SMS is much cheaper than making a phone call. So he always sends a mobile SMS instead of making a phone call. Currently, his mobile phone display has got a problem that it shows nothing on the screen. It is difficult for him to write a mobile SMS. But he knows the number arrangement of his mobile keypad. Now he asks your help to find out the text he has keyed in while writing the SMS. He will let you which key he has pressed and how many times. The keypad arrangement of his mobile is given in the following picture; '_' represents the space.



Input

The input file consists of several test cases. The first line of the input file contains a single integer **T<1000** indicating the number of test cases. Then **T** test cases follow. Each test case starts with a positive integer $\mathbf{5 \le L \le 100}$, which is the length of the message. Each of the next **two** lines contains **L** positive integers. First line contains $\mathbf{0 \le N_i \le 9}$, $\mathbf{1 \le i \le L}$ and second line contains $\mathbf{1 \le P_i \le 4}$, $\mathbf{1 \le i \le L}$. $\mathbf{N_i}$ are the keypad numbers which he types and $\mathbf{P_i}$ represents how many times he presses the $\mathbf{N_i}$ key.

Output

For each set of input produce one line of output "**message**", where **message** indicates the desired mobile SMS. See the sample input output for further clarification.

Sample Input	Output for Sample Input
2 17 9 3 5 2 6 6 3 0 8 6 0 4 4 8 7 2 1 1 2 3 3 3 1 2 1 1 3 1 3 3 2 1 3 1 12 4 6 9 0 2 7 3 0 9 6 8 1 2 3 1 1 1 3 2 1 3 3 2 3	welcome to iiupc. how are you?

Problem Setter: Saifur Rahaman
Alternate Solution: Tanveer Ahsan