Of Palindromes and Double Strings [1 Pt]

The **palindrome** is a string that can be read the same way from left to right and from right to left. For example, strings "aaaaa", "1221", "bbaabb" are **palindromes**, however the string "chef" is **not a palindrome** because if we read it from right to left, we will obtain "fehc" that is not the same as "chef".

We call a string a "**double string**" if it has an even length and the first half of this string is equal to the second half of this string, for example "abab" is a **double string** because the first half "ab" is equal to the second half "ab", however the string "abba" is **not a double string** because the first half "ab" is not equal to the second half "ba". The empty string "" is a **double string**, and its length is **0**.

Chef doesn't like palindromes, however he likes "double strings". He often likes to change the order of letters in some palindrome and sometimes to remove some symbols from it. Now he wonders: if a **palindrome** of length **N** is given, what is the maximal possible number of characters in a "double string" that can be obtained by removing and changing the order of symbols in it?

Input

Several test cases are given.
The first line of the sample input contains an integer **T** - the number of test cases.
Then, **T** lines follow.
Each line consists of a single integer **N** - the length of a palindrome.

Output

For each test case output a single integer - answer to the problem.

Constraints

- 1<=**T**<=10000
- 1<=**N**<=1000000000

Sample Input:

```
2
2
4
```

Sample Output:

```
2
4
```

The Number Game [1 Pt]

Alice and Bob play the following game. They choose a number **N** to play with. The rules are as follows:

1) Alice plays first, and the two players alternate.
2) In his/her turn, a player can subtract from **N** any proper divisor (not equal to **N**) of **N**. The number thus obtained is the new **N**.
3) The person who cannot make a move in his/her turn loses the game.

Assuming both play optimally, who wins the game?

Input

The first line contains the number of test cases T. Each of the next T lines contains an integer N.

Output

Output T lines, one for each test case, containing "ALICE" if Alice wins the game, or "BOB" otherwise.

Constraints

1 <= T <= 10000

1 <= N <= 1000000000

Sample Input:

```
2
1
2
```

Sample Output:

```
BOB
ALICE
```

For the first test case, Alice cannot make any move and hence Bob wins the game. For the second test case, Alice subtracts 1 from N. Now, Bob cannot make a move and loses the game.

Pie Cooling [1 Pt]

You have just finished baking several pies, and it's time to place them on cooling racks. You have exactly as many cooling racks as pies. Each cooling rack can only hold one pie, and each pie takes up an entire cooling rack, but you're not confident that the cooling racks can support the weight of the pies. You know the weight of each pie, and has assigned each cooling rack a maximum weight limit. What is the maximum number of pies that can cool on the racks?

Input:

Input begins with an integer **T**≤30, the number of test cases. Each test case consists of 3 lines. The first line of each test case contains a positive integer **N**≤30, the number of pies (and also the number of racks). The second and third lines each contain exactly positive **N** integers ≤ 100. The integers on the second line are the weights of the pies, and the integers on the third line are the weight limits of the cooling racks.

Output:

For each test case, output on a line the maximum number of pies that can placed on the racks.

Sample input:

```
2
3
10 30 20
30 10 20
5
9 7 16 4 8
8 3 14 10 10
```

Sample output:

```
3
4
```

Nuclear Pinball [1 Pt]

There are **K** nuclear reactor chambers labelled from 0 to **K-1**. Particles are bombarded onto Chamber 0. The particles keep collecting in the Chamber 0. However if at any time, there are more than **N** particles in a chamber, a reaction will cause 1 particle to move to the immediate next chamber (if current chamber is 0, then to chamber number 1), and all the particles in the current chamber will be destroyed and same continues till no chamber has number of particles greater than **N**. Given **K**, **N** and the total number of particles bombarded (**A**), find the final distribution of particles in the **K** chambers. Particles are bombarded one at a time. After one particle is bombarded, the set of reactions, as described, take place. After all reactions are over, the next particle is bombarded. If a particle is going out from the last chamber, it has nowhere to go and is lost.

Input

The input will consist of one line containing three numbers **A**, **N** and **K** separated by spaces. $0 \le \mathbf{A} \le 1000000000$, $0 \le \mathbf{N} \le 100$, $1 \le \mathbf{K} \le 100$ inclusive, all chambers start off with zero particles initially.

Output

Consists of K numbers on one line followed by a newline. The first number is the number of particles in Chamber 0, the second number is the number of particles in Chamber 1 and so on.

Sample Input:

```
3 1 3
```

Sample Output:

```
1 1 0
```

A total of 3 particles are bombarded. After particle 1 is bombarded, the chambers have particle distribution as "1 0 0". After second particle is bombarded, number of particles in Chamber 0 becomes 2 which is greater than 1. So, number of particles in Chamber 0 becomes 0 and in Chamber 1 becomes 1. So now distribution is "0 1 0". After the 3rd particle is bombarded, Chamber 0 gets 1 particle and so distribution is "1 1 0" after all particles are bombarded one by one.

Birthday Candles [1 Pt]

You're preparing a birthday cake for one of your friends, and you want to write his/her age in candles on the cake. There are 10 types of candles, one for each of the digits '0' through '9'. But you forgot the age of your friend, and don't know whether you have enough candles of the right types. For example, if your friend were 101 years old, you would need two '1' candles and one '0' candle. Given the candles you have, determine the smallest positive integer that cannot be represented with those candles.

Input:

Input will begin with an integer **T**≤100, the number of test cases. Each test case consists of a single line with exactly 10 integers, from 0 to 9, inclusive. The first integer of each test case represents the number of '0' candles, the second integer represents the number of '1' candles, and so on.

Output:

For each test case, output on a single line the smallest positive integer that cannot be expressed with the given candles.

Sample input:

```
3
2 1 1 4 0 6 3 2 2 2
0 1 1 1 1 1 1 1 1 1
2 2 1 2 1 1 3 1 1 1
```

Sample output:

```
4
10
22
```

Weirdest Bank Ever [2 Pts]

In Technopolis they have a very strange monetary system. Each Tech credit has an integer number written on it. A credit **n** can be exchanged at a bank for three smaller credits: **n/2**, **n/3** and **n/4**. But these numbers are all rounded down (the banks have to make a profit).

You can also sell Tech credits for American dollars; the exchange rate is 1:1. But since Technopolis has such a terrible economy, you can't buy Technopolis credits.

You have one credit of value **n**. What is the maximum amount of American dollars you can get for it?

Input

The input will contain ≤10 test cases. Each test case is a single line with a number 0≤**n**≤1000000000. That is the number written on your credit.

Output

For each test case output a single line, containing the maximum amount of American dollars you can make.

Sample Input:

```
12
2
```

Output:

```
13
2
```

You can change 12 into 6, 4 and 3, and then change these into $6+$4+$3 = $13.
If you try changing the 2 into 3 smaller coins, you will get 1, 0 and 0, and later you can get no more than $1 out of them. It is better just to change the 2 coin directly into $2.

Marble Grab [2 Pts]

Rohit dreams he is in a shop with an infinite amount of marbles. He is allowed to select **n** marbles. There are marbles of **k** different colors. From each color there are also infinitely many marbles. Rohit wants to have at least one marble of each color, but still there are a lot of possibilities for the rest of his selection. In his effort to make a decision he wakes up. Now he asks you how many possibilities for his selection he would have had. Assume that marbles of equal color can't be distinguished, and the order of the marbles is irrelevant.

Input

The first line of input contains a number **T** <= 100 that indicates the number of test cases to follow. Each test case consists of one line containing n and k, where **n** is the number of marbles Rohit selects and **k** is the number of different colors of the marbles. You can assume that $1 \leq k \leq n \leq 1000000$.

Output

For each test case print the number of possibilities that Rohit would have had. You can assume that this number fits into a signed 64 bit integer.

Sample Input:

```
2
10 10
30 7
```

Sample Output:

```
1
475020
```

Smoke Bombs! [2 Pts]

The Weasley twins are inventing a new smoke bomb! They have **n** mixtures, arranged in a row. Each mixture has one of 100 different colors (numbered from 0 to 99), and they need to mix all them together to produce the maximum amount of smoke. At each step, they take two mixtures that stand next to each other and mix them together, and put the resulting mixture back in place.

When mixing two mixtures of colors **a** and **b**, the resulting mixture will have the color (**a+b**) mod 100. The amount of smoke generated when mixing two mixtures of colors **a** and **b** is **a*b**. Find out what is the maximum amount of smoke that the twins can get when mixing all the mixtures together.

Input

There will be a number of test cases in the input.

The first line of each test case will contain n, the number of mixtures, 1 <= n <= 100.

The second line will contain n integers between 0 and 99 - the initial colors of the mixtures.

Output

For each test case, output the maximum amount of smoke.

Sample Input:

```
2
18 19
3
40 60 20
```

Sample Output:

```
342
2400
```

In the second test case, there are two possibilities:

- first mix 40 and 60 (smoke: 2400), getting 0, then mix 0 and 20 (smoke: 0); total amount of smoke is 2400
- first mix 60 and 20 (smoke: 1200), getting 80, then mix 40 and 80 (smoke: 3200); total amount of smoke is 4400

The second scenario is the correct approach since it maximizes the amount of smoke produced.

Not Towers of Hanoi [2 Pts]

Given **n** numbers, you can perform the following operation any number of times: Choose any subset of the numbers (possibly empty), none of which are 0. Decrement the numbers in the subset by 1, and increment the numbers not in the subset by **K**.

Is it possible to perform operations such that exactly **n-1** numbers become 0?

Input

The first line contains the number of test cases $1 \le$ **T** ≤ 1000. **2*T** lines follow, 2 for each case. The first line of a test case contains the numbers $2 \le$ **n** ≤ 100 and $1 \le$ **K** ≤ 10. The next line contains **n** numbers $a_1 \ldots a_n$, where $0 \le a_1 \le 1000$

Output

Output **T** lines, one corresponding to each test case. For a test case, output "YES" if there is a sequence of operations as described, and "NO" otherwise.

Sample Input

```
3
2 1
10 10
3 2
1 2 2
3 2
1 2 3
```

Sample Output

```
YES
YES
NO
```

Finding your roots [3 Pts]

You are given a binary tree consisting of 1 or more nodes. Each node has a unique integer ID. Each node has up to 2 children, which are identified by their IDs, and each node is the child of at most 1 other node. A node **X** is considered to be an ancestor of node **Y** if node **Y** is a child of node **X** or if there is some node **Z** for which **X** is an ancestor of **Z** and **Y** is a child of **Z**. No node is an ancestor of itself. A special node called the root node is an ancestor of all other nodes.

Unfortunately, you have forgotten which node is the root and your knowledge of the tree is incomplete. All you remember is the sum of the IDs of the children of each node.

Input

Input begins with an integer 1 ≤ **T** ≤ 50, the number of test cases. Each test case begins with an integer 1 ≤ **N** ≤ 30, the number of nodes in the tree. **N** lines follow with 2 integers each: the ID of a node, and the sum of the IDs of its children. Note that all node IDs are between 1 and 1000, inclusive. The second number will be 0 if the node has no children.

Output

For each test case, output on a line a space separated list of all possible values for the ID of the root node in increasing order. **It is guaranteed that at least one such ID exists for each test case.**

Sample Input

```
2                6           6
1                 \         / \
4 0                5       1   4
6                 / \       \
1 5              1   4       5
2 0             / \         / \
3 0            2   3       2   3
4 0
5 5
6 5
```

Sample Output

```
4
6
```

Explanation

In the first sample test case, there is only one node, which is clearly the root. In the second test case, there are two non-isomorphic trees that satisfy the constraints, as seen in the following picture:

Arithmetic Progressive [3 Pts]

Given **N** integers $\mathbf{A_1, A_2, \ldots. A_N}$, Dexter wants to know how many ways he can choose three numbers such that they are three consecutive terms of an arithmetic progression.

Meaning that, how many triplets **(i, j, k)** are there such that $\mathbf{1 \le i < j < k \le N}$ and $\mathbf{A_j - A_i = A_k - A_j}$.

So the triplets (2, 5, 8), (10, 8, 6), (3, 3, 3) are valid as they are three consecutive terms of an arithmetic progression. But the triplets (2, 5, 7), (10, 6, 8) are not.

Input

First line of the input contains an integer **N** $\mathbf{(3 \le N \le 100000)}$. Then the following line contains **N** space separated integers $\mathbf{A_1, A_2, \ldots, A_N}$ and they have values between **1** and **30000** (inclusive).

Output

Output the number of ways to choose a triplet such that they are three consecutive terms of an arithmetic progression.

Sample Input:

```
10
3 5 3 6 3 4 10 4 5 2
```

Output:

```
9
```

Explanation

The followings are all 9 ways to choose a triplet

```
1 : (i, j, k) = (1, 3, 5), (Ai, Aj, Ak) = (3, 3, 3)
2 : (i, j, k) = (1, 6, 9), (Ai, Aj, Ak) = (3, 4, 5)
3 : (i, j, k) = (1, 8, 9), (Ai, Aj, Ak) = (3, 4, 5)
4 : (i, j, k) = (3, 6, 9), (Ai, Aj, Ak) = (3, 4, 5)
5 : (i, j, k) = (3, 8, 9), (Ai, Aj, Ak) = (3, 4, 5)
6 : (i, j, k) = (4, 6, 10), (Ai, Aj, Ak) = (6, 4, 2)
7 : (i, j, k) = (4, 8, 10), (Ai, Aj, Ak) = (6, 4, 2)
8 : (i, j, k) = (5, 6, 9), (Ai, Aj, Ak) = (3, 4, 5)
9 : (i, j, k) = (5, 8, 9), (Ai, Aj, Ak) = (3, 4, 5)
```

Tiling Game [3 Pts]

You have a grid with 2 rows and **N** columns. Each square in the grid contains an integer **A**. You are given a supply of rectangular 2×1 tiles, each of which exactly covers two adjacent squares of the grid. You have to place tiles to cover all the squares in the grid such that each tile covers two squares and no pair of tiles overlaps.

The score for a tile is the difference between the bigger and the smaller number that are covered by the tile. The aim of the game is to maximize the sum of the scores of all the tiles.

Input

The first line contains one integer **N**, the number of columns in the grid. This is followed by 2 lines describing the grid, where $1 \leq \mathbf{A} \leq 10^4$. Each of these lines consists of $1 \leq \mathbf{N} \leq 10^5$ integers, separated by spaces.

Output

A single integer indicating the maximum score that can be achieved by any tiling of the given grid.

Sample Input
4
8 6 2 3
9 7 1 2
Sample Output
12

Here is an example of a grid, along with two different tilings and their scores. The score for Tiling 1 is $12 = (9 - 8) + (6 - 2) + (7 - 1) + (3 - 2)$ while the score for Tiling 2 is $6 = (8 - 6) + (9 - 7) + (3 - 2) + (2 - 1)$. There are other tilings possible for this grid, but you can check that Tiling 1 has the maximum score among all tilings. Your task is to read the grid of numbers and compute the maximum score that can be achieved by any tiling of the grid.

Prime Distance on a Tree [5 Pts]

You are given a tree. If we select 2 distinct nodes uniformly at random, what's the probability that the distance between these 2 nodes is a prime number?

Input

The first line contains a number $2 \leq \mathbf{N} \leq 50{,}000$: the number of nodes in this tree.
The following **N**-1 lines contain pairs a[i] and b[i], which means there is an edge with length 1 between a[i] and b[i].

Output

Output a real number denoting the probability we want. We'll accept answers within 10^-6 error margin.

Sample Input:

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

Sample Output:

```
0.5
```

We have C(5, 2) = 10 choices, and these 5 of them have a prime distance:

1-3, 2-4, 3-5: 2

1-4, 2-5: 3

Note that 1 is not a prime number.

Graph Queries [5 Pts]

You are given an undirected graph **G** consisting of **N** vertices and **M** edges. Each vertex has a unique index from **1** to **N**, and each edge has a unique index from **1** to **M**.

You also have **Q** pairs of integers: $\mathbf{L_i}$, $\mathbf{R_i}$ ($\mathbf{1 \le L_i \le R_i \le M}$). For each pair $\mathbf{L_i}$, $\mathbf{R_i}$, you want to know: how many connected components will contain graph **G** if you erase all the edges from the graph, except the edges with indies **X**, where $\mathbf{L_i \le X \le R_i}$.

Input

The first line of the input contains an integer $\mathbf{1 \le T \le 1000}$ denoting the number of test cases. The first line of each test case contains three integers $\mathbf{1 \le N, M, Q \le 200000}$. Each of the next **M** lines contains a pair of integers $\mathbf{1 \le U_i, V_i \le N}$, the current edge of graph **G**. Each of the next **Q** lines contains a pair of integers $\mathbf{1 \le L_i \le R_i \le M}$, the current query.

Output

For each query of each test case print the required number of connected components.

Constraints

- Sum of all values of **N** for test cases is not greater than **200000**. Sum of all values of **M** for test cases is not greater than **200000**. Sum of all values of **Q** for test cases is not greater than **200000**.
- Graph **G** can contain self-loops and multiple edges.

Sample Input:

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

Output:

```
2
1
3
1
1
```