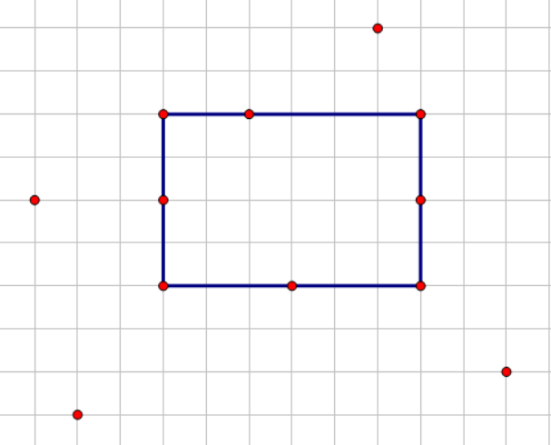
**A. rectangle**

**Problem Description**

There are n points in a geometrical plane. If there exists a rectangle whose four vertices are among those points and every edge of the rectangle has exactly m points which are among those points, now ask how many rectangles existing in this plane could satisfy all the conditions. The edge of the rectangle should be parallel to the coordinate axes. This is a rectangle that satisfy the conditions when m=3 represented in the chart below.



**Input**

First line contains T (T ≤20) denoting the number of test cases.

T cases follows for each cases:

First line contains two integers n, m (5≤ n ≤100000, m < n)

Followed by n lines, each line contains two integers xi, yi indicates the location of the ith point. There are no two points at the same location. (|Xi|, |Yi|≤10000000)

**Output**

For each case, output an integer indicate the number of rectangles that satisfy the condition.

**Sample Input**

1

9 2

0 0

0 3

0 6

3 0

3 3

3 6

6 0

6 3

6 6

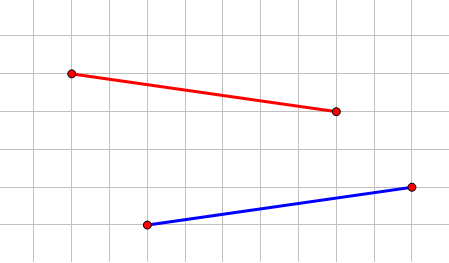
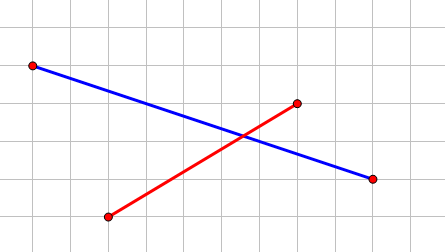
**Sample Output**

4

**B. Travel**

**Problem Description**

There are **N** points on a geometrical plane. Doctor.Gu start at the leftmost point and Doctor.Jia start at the second left point.Then they can continually choose a point which is on the right of the point they stay to go on the journey, until they have no point to choose.It is requested that all the points should be visited but only once(can’t be visited by both Doctor.Gu and Doctor.Jia). And their route can’t be crossed. Please output the number of different plans **mod 1e9+7**.



**Input**

First line contains T(T ≤ 20) denoting the number of test cases.

T cases follows. For each cases:

First line contains a number **n** indicates the number of points(2 < n ≤ 1000)

Next **n** lines, contains two integers **xi,** **yi,** indicates the **ith** point. The n points is sorted by x and there are no three points collinear, and no two points that xi = xj, (i  j).

**Output**

For each case, output the number of plans mod 1e9 + 7.

**Sample input**

1

5

-1 3

1 4

2 5

3 1

5 2

**Sample output**

6

**Hint**

Dr.Gu:1, Dr.Jia:2->3->4->5; Dr.Gu:1->3->4->5, Dr.Jia:2;

Dr.Gu:1->3->5, Dr.Jia:2->4; Dr.Gu:1->3, Dr.Jia:2->4->5;

Dr.Gu:1->4->5, Dr.Jia:2->3; Dr.Gu:1->4, Dr.Jia:2->3->5;

Six plans in total.

**C.Circle**

**Problem Description**

There are n points in a geometrical plane, ask if there is a circle that contains at least ⌊n/3⌋ points on it or not. The point is on the circle (center: (x0, y0), radius: r) if (x - x0)^2 + (y - y0)^2 = r^2.

**Input**

First line contains T (T≤20) denoting the number of test cases.

T cases follows. For each cases:

First line contains an integer n (3 ≤ n ≤ 30000), indicates the number of the points.

Followed by n lines, each line contains two numbers Xi, Yi, indicates the location of the ith point. Xi, Yi retain six decimal fractions.

(|Xi|, |Yi| ≤ 10000000)

**Output**

For each case, if there exists a circle meet the condition, output "YOUGE", otherwise, output "NIUGE"

**Sample Input**

2

5

1.213551 0.151532

1.515114 1.451512

2.566665 1.531351

3.516151 5.561565

6.515162 8.515195

10

1.000000 0.000000

2.000000 0.000000

3.000000 0.000000

4.000000 0.000000

5.000000 0.000000

6.000000 0.000000

7.000000 0.000000

8.000000 0.000000

9.000000 0.000000

10.000000 0.000000

**Sample Output**

YOUGE

NIUGE

**D. Delete**

**Problem Description**

Definite an operation work(Q,x) as sequence Q from the first number on,keep the top x numbers and delete followed x numbers, keep the next x numbers and delete followed x numbers, then keep recycling this process.

Q is the natural number sequence, Given a sequence A, operating Work(Q,Ai) upon Natural number sequence output the nth member after n operations.

**Input**

First line contains T (T ≤ 10) denoting the number of test cases.

T cases follows. For each cases:

First line contains integer n (n ≤ 10000), next line contains n integers Ai (0 < Ai ≤ 1000) indicates the ith operation.

**Output**

For each case output the answer in one line.

**Sample Input**

2

3

1 2 3

3

3 2 1

**Sample Output**

9

15

**E. Embrace changes**

**Problem Description**

There is a company with N employees and each employee has an ID from 1 to N. Everyone has a direct leadership except No.1（The one’s leadership has a smaller ID number than he). Everyone has a quantity of work Ai, and everyone has his rank, The No.1 rank is 1, other people’s rank is his direct leadership’s rank plus 1. In this problem, we think everyone leads himself.

Now, there are some changes in this company:

1. Ask the sum of work quantities who is under No.X’s leader and the rank between [L, R].

2. Add y to the work quantity who is under the No.X’s leader and the rank between [L, R].

3. Change the work quantity to y who is under the No.X’s leader and the rank between [L, R].

4. Ask the sum of work quantities that No.x and all his leaderships.

5. Add y to the work quantity that No.x and all his leaderships.

6. Change the work quantity to y that No.x and all his leaderships.

1 ≤ x, l, r ≤ n, |y| ≤ 1000

At first, everyone’s work quantity is 0.

There may be no employees between [L, R].

**Input**

First line contains T (T ≤ 10) denoting the number of test cases.

T cases follows. For each cases:

First line contains two numbers n, m (1 ≤ n, m ≤ 100000) indicates the number of nodes and the number of operations.

Next lines contains n - 1 integers indicating the direct leadership of No.2 to No.N employees.

Next m lines, each line indicate an operation.

First integer is the type of operations:

If type = 1, followed by 3 integers indicating x, L, R

If type = 2, followed by 4 integers indicating x, L, R, y

If type = 3, followed by 4 integers indicating x, L, R, y

If type = 4, followed by an integer indicating x

If type = 5 followed by 2 integers indicating x, y.

If type = 6 followed by 2 integers indicating x, y

**Output**

For operation 1 and operation 4， output one number indicate the sum.

**Sample Input**

1

5 6

1 1 2 2

2 1 1 2 1

1 1 1 1

3 2 2 4 1

5 5 1

6 3 2

4 5

**Sample Output**

1

6

**F. Personal Income Tax**

**Problem Description**

Personal income tax is to adjust the tax laws and regulations of general social relations between the authorities and natural persons (residents and non-residents) and management of personal income tax levied in the course of what happened.

Personal income taxpayers, both residents of the taxpayer, including non-resident taxpayer. Resident taxpayer obligations fully taxable and must be on its territory from China, all foreign income pay individual income tax; and not only in respect of resident taxpayers resulting from Chinese territory, to pay personal income tax.

Personal income tax is a tax on income countries, their citizens, resulting residing in the territory of foreign individuals and individuals from national tax. In some countries, the personal income tax is the main tax, a larger proportion of fiscal revenue, also have a greater impact on the economy.

**3500 yuan exemption amount**

|  |  |
| --- | --- |
| Taxable income | taxrate % |
| ≤ 1500 | 3 |
| 1500< x ≤ 4500 | 10 |
| 4500< x ≤ 9000 | 20 |
| 9000< x ≤ 35,000 | 25 |
| 35,000< x ≤ 55,000 | 30 |
| 55,000< x ≤ 80,000 | 35 |
| > 80, 000 | 45 |

**Taxable income = total income - exemption amount**

**Input**

First line contains T (T ≤ 10000) denoting the number of test cases.

T cases follows. For each cases:

There is an integer **n (0 < n ≤ 10 ^ 6)** indicates the **total income** of Dr.Gu.

**Output**

For each case output the tax Dr.Gu need to pay. (The answer should be reserved two decimal fractions)

**Sample Input**

3

10000

3000

33333

**Sample Output**

745.00

0.00

6453.25

**Hint**

For the first sample, the total income is 10000, subtract 3500 yuan exemption amount, so the taxable income is 6500 yuan. So the answer is 1500 \* 3% + (4500 - 1500) \* 10% + (6500 - 4500) \* 20 = 745.00 yuan.

**G. Chinese Mahjong**

**Problem Description**

Mahjong is a game of Chinese origin usually played by four persons with tiles resembling dominoes and bearing various designs, which are drawn and discarded until one player wins with a hand of four combinations of three tiles each and a pair of matching tiles.

A set of Mahjong tiles will usually differ from place to place. It usually has at least 136 tiles, most commonly 144, although sets originating from America or Japan will have more. The 136-tile Mahjong includes:

**Dots**: named as each tile consists of a number of circles. Each circle is said to represent copper (tong) coins with a square hole in the middle. In this problem, they’re represented by T1, T2, T3, T4, T5, T6, T7, T8 and T9.



**Bams**: named as each tile (except the 1 Bamboo) consists of a number of bamboo sticks. Each stick is said to represent a string (suo) that holds a hundred coins. In this problem, they’re represented by S1, S2, S3, S4, S5, S6, S7, S8 and S9.



**Craks**: named as each tile represents ten thousand (wan) coins, or one hundred strings of one hundred coins. In this problem, they’re represented by W1, W2, W3, W4, W5, W6, W7, W8 and W9.



**Wind tiles**: East, South, West, and North. In this problem, they’re represented by Dong, Nan, Xi, Bei.



**Dragon tiles:** red, green, and white. The term dragon tile is a western convention introduced by Joseph Park Babcock in his 1920 book introducing Mahjong to America. Originally, these tiles are said to have something to do with the Chinese Imperial Examination. The red tile means you pass the examination and thus will be appointed a government official. The green tile means, consequently you will become financially well off. The white tile (a clean board) means since you are now doing well you should act like a good, incorrupt official. In this problem, they’re represented by Zhong, Fa, Bai.



There are 9 ∗ 3 + 4 + 3 = 34 kinds, with exactly 4 tiles of each kind, so there are 136 tiles in total.

Chinese Mahjong is very complicated. However, we only need to know very few of the rules in order to solve this problem. A meld is a certain set of tiles in one’s hand. There are three kinds of melds you need to know:

**Pong**: A set of three identical titles.

Example:

，



**Chow**: A set of three suited tiles in sequence. All three tiles must be of the same suites. Sequences of higher length are not permissible (unless it forms more than one meld). Obviously, wind tiles and dragon tiles can never be involved in chows.

Example:

，



**Eye**: The pair, while not a meld, is the final component to the standard hand. It consists of any two identical tiles.

Example:

，



A player wins the round by creating a standard mahjong hand. That means, the hand consists of an eye and several (possible zero) pongs and chows. Note that each title can be involved in exactly one eye/pong/chow.

When a hand is one tile short of wining, the hand is said to be a ready hand, or more figuratively, ’on the pot’. The player holding a ready hand is said to be waiting for certain tiles. For example is waiting for , or 。



These tiles which can make the players win in the round is called **First tiles.** Of course, there are also **Second tiles:** the second tiles is the tile that we can catch this tile and drop another tile to make the number of First tiles increase.

Now, give you 13 tiles, please calculate the number of **Second tiles.**

**In this problem, we don’t consider the influence of other players.**

**Input**

First line contains T (T ≤ 10) denoting the number of test cases.

Each case consists of 13 tiles in a single line. The hand is legal (e.g. no invalid tiles, exactly 13 tiles). There is a space between every two tile.

**Output**

For each case, print the number of **Second tiles**.

**Sample Input**

2

T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 Dong Dong Nan

W1 W2 W3 W4 W5 W6 Dong Nan Xi Bei Zhong Fa Bai

**Sample Output**

7

0

**Hint**

For the first sample, the number of First tiles is 0, and there are 7 types of tiles can be the **Second tiles**:



For the second sample, there is none **Second tiles**.

**H. work group**

**Problem Description**

A group is made of n people who are numbered from 0 to n-1. Now they are requested to complete a project, and there should be only one member working on every stage while others take rest. For number i, if he is working on the first stage, he would transfer the next stage to (i + Ai) % n or (i - Ai) % n after finishing this stage. If the previous stage was worked by someone whose number is even number, he would transfer the next stage to number (i + Ai) % n after finishing this stage, if he is not working on the first stage or the previous stage is completed by someone whose number is odd number, he would transfer the next stage to number (i – Ai + n) %n after finishing this stage.

Now ask who will work on only one stage at most.(No matter who works on the first stage and no matter the first worker handover the next stage to (i + Ai) % n or (i - Ai) % n, they will work on only one stage at most.)

**Input**

First line contains T (T ≤ 10) denoting the number of test cases.

T cases follows. For each cases:

First line contains n (1 ≤ n ≤ 100000)

Next line contains n numbers Ai (|Ai| < n)

**Output**

For each case, output two lines.

First line contains a number indicates the quantity of members who satisfy the request.

Next line output the number of each member who satisfy the request.

If no one satisfy the request, the second line is blank.

**Sample Input**

1

5

1 2 3 1 2

**Sample Output**

1

0

**I. Fatigue Legend**

**Problem Description**

*HearthStone: Heroes of Warcraft* is a very popular game developed by Blizzard. This problem is based on this game, but even if you don’t know this game, you can solve this problem easily.

Each Hearthstone battle is a 1 vs 1 match between two opponents. Gameplay in Hearthstone is turn-based, with players taking turns to play cards from their hands.

Each player can choose a '**hero**', an important character from Warcraft. Each hero is associated with a unique hero power. Each hero has 30 Health, and if their Health is reduced to zero, the hero is destroyed and the controlling player loses the game.

At the start of each turn, the player draws a new card from their deck —— a collection of 30 cards selected before battle. During their turn, each player can choose to play any of their cards or use the hero power. However, these actions require the player to spend mana and this limitation forces players to strategically plan out their moves. Each player starts the game with 1 Mana Crystal, and gains one more at the start of each turn until they reach the maximum of 10 Mana Crystals.

However, once a player has drawn all of their cards, attempts to draw an additional card from their empty deck will cause them to suffer damage from Fatigue. Fatigue initially deals 1 damage to the player, but this amount increases by 1 each time.

Now, consider this condition: the cards in their hands and deck has been used out, of course, the two players have 10 Mana Crystals. That means, in each turn, the player only has two operation:

(1) Draw a card from their empty deck

(2) Use their hero power (the hero power cost 2 mana and can only be used once at most in each turn)

Definite n is the Health of the hero, the Health can’t excess 30.

Definite m is the Armor of the hero, the hero’s Health is reduced only if the Armor has been reduced to 0. The Armor of the hero can increase and no upper limitation.

To simplify this problem, give you four heroes to choose:

|  |  |  |  |
| --- | --- | --- | --- |
| hero id | hero | hero power | description of hero power |
| 1 | Jaina | Fire Blast | deal 1 damage to any target |
| 2 | Rexxar | Steady Shot | dealing 2 damage direct to the enemy hero |
| 3 | Garrosh | Armor Up | Gain 2 Armor |
| 4 | Anduin | Lesser Heal | Restore 2 Health |

Give you the hero id, hero Health, hero Armor of the two players, whether the first player can win this game?(the two players haven’t suffered damage from Fatigue)

**Input**

First line contains T (T100) denoting the number of test cases.

T cases follow. For each case:

First line contains three integers X1, N1, M1, represent the first player’s hero id, hero Health, hero Armor.

Second line contains three integers, X2, N2, M2 represent the second player’s hero id, hero Health, hero Armor.

(1≤X1,X2<5; 1≤N1,N2≤30; 0≤M1,M2≤100)

**Output**

For each case, if the first player can win the game, output "YES", otherwise, output "NO".

**Sample Input**

2

1 10 3

4 5 10

3 10 10

1 10 10

**Sample Output**

NO

YES

**J.S number**

**Problem Description**

S number is the number which the sum of every digit is a prime number, such as number 98, 29.Output the number of S number in[L,R].

**Input**

First line contains T (T ≤ 10) denoting the number of test cases.

T cases follows. For each cases:

There two numbers L, R. (0 ≤ L ≤ R ≤ 10^16)

**Output**

For each case, output the number of S number.

**Sample Input**

2

3

1 2 3

3

3 2 1

**Sample Output**

9

15