The 3rd Universal Cup

Stage 7: Warsaw

August 24-25, 2024

This problem set should contain 13 problems (A to M) on 17 numbered pages.

Based on

amppz

Akademickie Mistrzostwa Polski w Programowaniu Zespoowym (AMPPZ)

Hosted by

UNIVERSITAS VARSOVIENSIS

Problem A. Bus Analysis

Time limit:	2 seconds
Memory limit:	1024 megabytes

Adam is embarking on a very long journey by public bus. He has the option to purchase 20-minute tickets for 2 PLN and 75-minute tickets for 6 PLN[1]. Tickets can be purchased at any whole minute, and they are automatically validated. For example, a 20-minute ticket purchased at 9:14:00 is valid until 9:33:59. Adam can purchase a new ticket even if he still has a valid one.

The bus stops at N stops, i-th stop at minute t_i. At each stop, ticket inspectors may be waiting. They board the bus, immediately check all passengers, and then disembark at the same stop (this process takes only a few seconds).

At the very beginning of the journey, Adam will receive information about which stops have ticket inspectors waiting. He will then plan his ticket purchases to minimize his total expenses and have a valid ticket at the time of each inspection.

Consider all 2^N scenarios of ticket inspector placements and calculate Adam's optimal expenses in PLN. Output the remainder of the sum divided by $10^9 + 7$.

Input

The first line of the input contains an integer N ($1 \le N \le 1000$) – the number of stops.

The second line contains an increasing sequence of N integers $t_1, t_2, \ldots, t_N$ ($1 \le t_i \le 10^9$, $t_i < t_{i+1}$) – the minutes at which the bus stops.

Output

Output a single integer – the remainder of the result divided by $10^9 + 7$.

Examples

standard input	standard output
3 1 8 20	14
5 25 45 65 85 1000000000	156

Note

In the first sample test case there are 3 stops and 8 scenarios to consider. If Adam receives information that no ticket inspector is present at any of the stops, he won't buy any tickets. In the remaining 7 scenarios, he only needs one 20-minute ticket costing 2 PLN. The answer is $0 + 7 \cdot 2 = 14$ PLN.

Let's consider 2 out of 32 scenarios in the second sample test case:

- Ticket inspectors at all 5 stops ($25, 45, 65, 85, 1\,000\,000\,000$) – Adam spends 8 PLN: 6 PLN on a 75-minute ticket and 2 PLN on a 20-minute ticket, for example bought at minutes 25 and 1 000 000 000.
- Ticket inspectors at 2 stops ($25, 45$) – Adam spends 4 PLN on two 20-minute tickets, for example bought at minutes 25 and 45 or minutes 17 and 26.

[1]The actual prices in Warsaw are 3.40 PLN for 20 minutes and 4.40 PLN for 75 minutes or the entire route. It's usually optimal to buy longer tickets.

Problem B. Missing Boundaries

Time limit: 4 seconds
Memory limit: 1024 megabytes

An interval of integers $[1, L]$ was split into N non-empty intervals of integers $[a_i, b_i]$ ($1 \le a_i \le b_i \le L$, for $1 \le i \le N$) in such a way that every integer in $[1, L]$ belongs to exactly one of the intervals $[a_i, b_i]$. Then, some of the boundaries of the obtained intervals $[a_i, b_i]$ were hidden.

You are given a set of intervals in which some boundaries may be missing. Your task is to determine if they could have been produced by the method described above, meaning that it is possible to replace the missing values in such a way that the intervals are non-empty, pairwise disjoint, and together they cover all integers from 1 to L.

Input

The first line of the input contains an integer T ($1 \le T \le 30\,000$) indicating the number of test cases. Then there are T descriptions of test cases, one after the other.

The first line of each individual test case description contains two integers N and L ($1 \le N \le 200\,000$, $1 \le L \le 10^9$), indicating respectively the number of given intervals and the length of the original interval that would be split into parts. Each of the following N lines contains two integers, a_i and b_i ($-1 \le a_i, b_i \le L$; $a_i, b_i \ne 0$). The number -1 denotes a missing value. If both numbers a_i and b_i are positive, then $a_i \le b_i$.

The sum of the numbers N in all test cases is not greater than $200\,000$.

Output

The output should contain exactly T lines. The i-th line should contain the answer to the i-th test case: the word TAK if the intervals described in the input could have been generated using the method described in the task, or the word NIE if it is not possible.

Example

standard input	standard output
3 4 51 1 -1 11 50 -1 -1 -1 10 3 2 -1 -1 -1 -1 -1 -1 2 3 1 2 2 3	TAK NIE NIE

Note

In the first test case, we have $L = 51$, and the interval $[1, 51]$ can be split, for example, into intervals $[1, 7]$, $[8, 10]$, $[11, 50]$, and $[51, 51]$. After hiding some values, these intervals match the given intervals $[1, -1]$, $[-1, 10]$, $[11, 50]$, and $[-1, -1]$. Therefore, the answer to this test case is TAK.

In the second test case, we have 3 intervals, all of which have all boundaries hidden. However, the interval $[1, 2]$ contains only two integers, so it is impossible to split it into 3 non-empty and disjoint intervals of integers. Therefore, the answer to this test case is NIE.

In the third test case, we have two intervals $[1, 2]$ and $[2, 3]$ with no missing boundaries. The number 2 is covered twice so the answer is `NIE`.

Problem C. Price Combo

Time limit: 2 seconds
Memory limit: 1024 megabytes

Czesaw has decided to build his own computer. After careful consideration, he has chosen an entire configuration and made a list of N components that he needs to purchase.

There are two electronics stores in the area: AutoAGD and BioBezpieczniki. The i-th component on Czesaw's list costs A_i PLN at the AutoAGD store and B_i PLN at the BioBezpieczniki store. Each of the stores independently offers a "Price Combo: Buy 1, Get 1 Free" promotion – when buying two products, the cheaper one is free (ties resolved in any way). The promotion is unlimited, meaning the same customer can make as many purchases in one store, each time taking advantage of the promotion.

Help Czesaw buy all the components at the lowest possible total cost. Output the minimum possible sum of his expenses.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 200\,000$), which represents the number of components Czesaw needs.

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^9$), representing the prices of the components at the AutoAGD store.

Similarly, the third line contains N integers $B_1, B_2, \ldots, B_N$ ($1 \leq B_i \leq 10^9$), representing the prices at the BioBezpieczniki store.

Output

Output a single integer – the lowest possible amount (in PLN) spent by Czesaw on the purchase of all N components.

Example

standard input	standard output
7 10 12 19 99 10 8 49 9 14 15 199 11 7 19	131

Note

Czesaw wants to buy 7 components. He can make the following purchases:

- In AutoAGD, he buys components 1 (10 PLN) and 5 (10 PLN), paying 10 PLN.
- In AutoAGD, he buys components 4 (99 PLN) and 7 (49 PLN), paying 99 PLN.
- In BioBezpieczniki, he buys components 2 (14 PLN) and 3 (15 PLN), paying 15 PLN.
- In BioBezpieczniki, he buys component 6 for 7 PLN.

All purchases would cost a total of $10 + 99 + 15 + 7 = 131$ PLN.

Problem D. Data Determination

Time limit: 2 seconds
Memory limit: 1024 megabytes

There was once a very serious scientist who decided to answer once and for all the Ultimate Question of Life, the Universe, and Everything. He began with theoretical considerations and ultimately concluded that the answer is a positive integer and equals m. However, these considerations were based on many uncertain assumptions regarding life, the universe, and everything. Theoretical considerations should be supported with experimental evidence!

The scientist designed a special experiment, burdened with various measurement errors. He conducted it n times, and the result of the i-th experiment is the number a_i. In his scientific work, he plans to include data from exactly k experiments, and their median[2] must be exactly m to confirm his theory.

Verify if he can achieve his goal. Write a program that, given the results of all n experiments, determines whether it is possible to select k of them in such a way that the median of their results is exactly m.

Input

The first line of the input contains a single integer t ($1 \le t \le 10\,000$), which represents the number of independent scenarios to consider. Each scenario is described in two lines.

In the first line of each scenario, there are three integers n, k and m ($1 \le k \le n \le 200\,000$, $1 \le m \le 10^9$), representing the number of conducted experiments, the number of experiments required for scientific work, and the desired median, respectively. The second line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$), representing the results of the experiments.

The sum of values n in all test cases does not exceed $200\,000$.

Output

The output should consist of t lines, containing answers for each scenario. In the i-th line, there should be one word `TAK` if it is possible to select the appropriate k experiments in the i-th scenario, or one word `NIE` otherwise.

Example

standard input	standard output
3 6 4 42 41 43 41 57 41 42 4 2 4 1 2 5 8 7 5 57 101 2 42 5 57 7 13	TAK NIE NIE

Note

In the first scenario, you can select experiments with results $(41, 43, 41, 57)$; after sorting, you get the sequence $(41, 41, 43, 57)$, where the arithmetic mean of the two middle elements is $\frac{43+41}{2} = 42$.

In the second scenario, it is not possible to select a pair of elements with a median of 4. For example:

- The sequence $(2, 5)$ has a median of $\frac{2+5}{2} = 3.5$, which is too low.
- On the other hand, the sequence $(1, 8)$ has a median of $\frac{1+8}{2} = 4.5$, which is too high.

[2]The *median* of a sequence is the middle element when sorted. If the length of the sequence is even, it is the arithmetic mean of the two middle elements. For example, the median of the sequence $(9, 7, 3, 4, 5)$ is 5, and the median of the sequence $(3, 1, 6, 6)$ is $\frac{3+6}{2} = 4.5$.

Problem E. Express Eviction

Time limit: 4 seconds
Memory limit: 1024 megabytes

Bajtocja is a rectangular land consisting of $H \cdot W$ cells arranged in H rows and W columns. The border of each cell is a local road with little traffic. Each cell is either empty or inhabited by one resident. The residents like peace and would not want to live right next to a planned express road[3].

Your task is to plan the route of the express road from the upper-left to the lower-right corner of Bajtocja. The route can only run along existing roads, which minimizes reconstruction costs. The length and number of turns do not matter.

It will be necessary to evict each resident whose cell is adjacent to the express road by the side or even by a corner. At least how many residents need to be evicted?

Input

The first line of the input contains two numbers H and W ($1 \leq H, W \leq 50$) – the dimensions of Bajtocja.

The next H lines describe inhabited and uninhabited cells. Each line contains a string of length W, consisting of characters '.' (empty cell) and '#' (inhabited cell).

Output

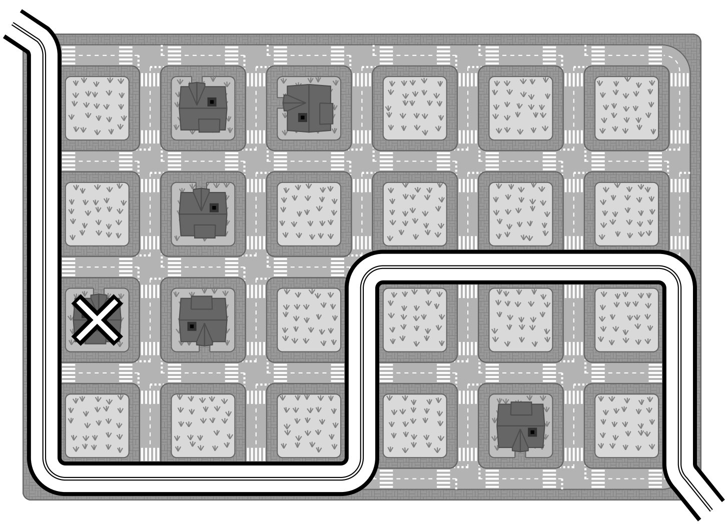
Output a single integer – the minimum possible number of evicted residents.

Example

standard input	standard output
4 6 .##... .#.... ##.... ....#.	1

Note

If you evict the resident from the cell in the first column and the third row, you can build the express road as follows:



[3]Many years ago, an old man was so annoyed that he tied many balloons to his house and flew away.

Problem F. Fibonacci Fusion

Time limit: 4 seconds
Memory limit: 1024 megabytes

The Fibonacci numbers are $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$ – each number (except the first two) is the sum of the two previous numbers.

You are given N numbers a_i. Count the pairs $i < j$ such that $a_i + a_j$ is a Fibonacci number.

Input

The first line of the input contains an integer N ($1 \leq N \leq 200\,000$).

Each of the next N lines contains an integer a_i ($1 \leq a_i < 10^{2\,000\,000}$). The total number of digits in these numbers does not exceed $5\,000\,000$.

Output

Output a single integer – the number of pairs that sum to a Fibonacci number.

Example

standard input	standard output
6 50 8 8 5 72 354224848179261915070	4

Note

There are 4 such pairs:

- $a_2 + a_4 = 8 + 5 = 13$
- $a_3 + a_4 = 8 + 5 = 13$
- $a_1 + a_4 = 50 + 5 = 55$
- $a_4 + a_6 = 5 + 354224848179261915070 = 354224848179261915075$

Problem G. Game of Geniuses

Time limit: 2 seconds
Memory limit: 1024 megabytes

Two geniuses are playing the following game. They have a square $n \times n$ board filled with integers. Players take turns making moves. The first player's move is to cross out a selected row that hasn't been crossed out yet. The second player's move is to cross out a selected column that hasn't been crossed out yet. After each player makes $n-1$ moves, exactly one integer will remain uncrossed. The first player aims to maximize this number, while the second player aims to minimize it.

Given a board, determine the value of the one uncrossed number at the end of the game. As the two players are geniuses, they play optimally.

Input

The first line of the input contains a single integer n ($2 \le n \le 50$), representing the size of the board. The following n lines of the input describe the rows of the board: the i-th line contains n integers $a_{i1}, a_{i2}, \ldots, a_{in}$ ($1 \le a_{ij} \le 2500$), representing the numbers in the i-th row.

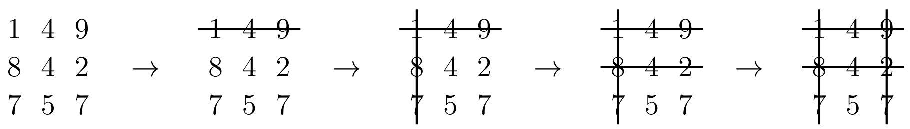
Output

Output a single integer, the value of the one uncrossed number at the end of the game when both players play optimally.

Example

standard input	standard output
3 1 4 9 8 4 2 7 5 7	5

Note



Problem H. Henry the Plumber

Time limit:	1 second
Memory limit:	1024 megabytes

Henryk has to complete a certain number of three-dimensional hydraulic orders. Each order consists of connecting the water inlet to the outlet. Such connections are made using a sequence of pipes and elbow fittings. Pipes are straight sections, and elbows connect the two ends of the pipes at right angles. At the water inlet, we must place the first elbow according to the specified direction of the inlet, and at the water outlet, we must place the last elbow according to the specified direction of the outlet.

Both the inlet and the outlet are located **horizontally**, meaning the vectors of these directions have a zero z-coordinate z.

For simplicity, we assume that the inlet, outlet, and all elbows have zero dimensions, meaning they are points in three-dimensional space $\mathbb{R}^3$, and the pipes have zero width, meaning they are segments in three-dimensional space $\mathbb{R}^3$.

The inlet and outlet are in different locations. The first pipe must start at the water inlet point and be perpendicular to the inlet direction. Each subsequent pipe must start at the end of the previous one and be perpendicular to it. The last pipe must end at the water outlet point and be perpendicular to the outlet direction. Each pipe must have a positive length.

Pipes cannot intersect at any other locations than those described above. In particular, all elbows must be at different points and cannot be inside any pipe, and pipes cannot share common points except for the elbows.

For each order, decide whether it can be completed, and if so, calculate the minimum number of elbows needed to complete it.

Input

The first line of the input contains the number of orders t ($1 \le t \le 100$). Each order is described in four lines.

In the first line of each order, there are three integers x_1, y_1, z_1 ($-20 \le x_1, y_1, z_1 \le 20$) describing the point (x_1, y_1, z_1) of the water inlet.

The second line of the order contains 2 integers p_1, q_1 ($-20 \le p_1, q_1 \le 20$, $(p_1, q_1) \neq (0, 0)$) describing the direction of the water inlet.

The next two lines describe the coordinates (x_2, y_2, z_2) and the direction of the water outlet (p_2, q_2) in the same format.

The water inlet and outlet are at different points: $(x_1, y_1, z_1) \neq (x_2, y_2, z_2)$.

The water flows horizontally along the vector $(p_1, q_1, 0)$, meaning from the direction of the point $(x_1 - p_1, y_1 - q_1, z_1)$. The outflow is also horizontal, along the vector $(p_2, q_2, 0)$, meaning in the direction of the point $(x_2 + p_2, y_2 + q_2, z_2)$.

The length of these vectors is not important; only their direction and orientation matter. The orientation is in the direction of water movement, i.e., towards the elbow in the inlet and away from the elbow in the outlet.

Output

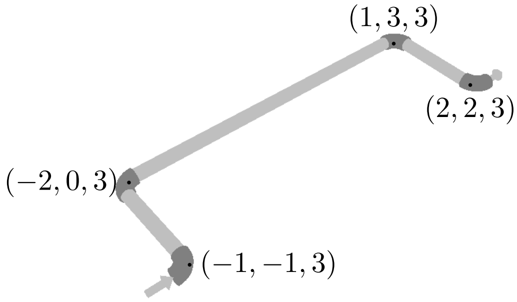
The output should consist of t lines. In the i-th line, you should provide the minimum number of elbows needed to complete the order or the word `NIE` if the order is not feasible.

Example

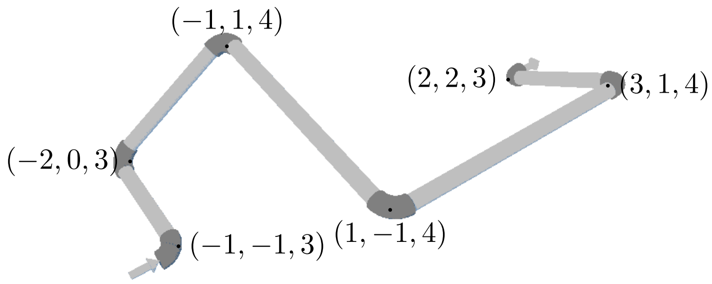
standard input	standard output
2 -1 -1 3 1 1 2 2 3 2 2 5 5 1 3 0 7 6 -2 1 -2	4 3

Note

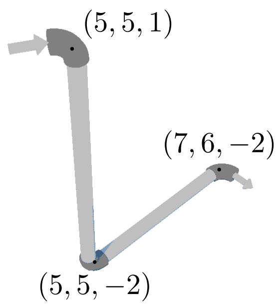
In the first example, the optimal solution is to use four elbows, for example, sequentially at the points $(-1,-1,3)$, $(-2,0,3)$, $(1,3,3)$, $(2,2,3)$:



An inefficient solution would be to use six elbows at points $(-1,-1,3)$, $(-2,0,3)$, $(-1,1,4)$, $(1,-1,4)$, $(3,1,4)$, $(2,2,3)$:



In the second example, we can use three elbows at points $(5,5,1)$, $(5,5,-2)$, $(7,6,-2)$:



Problem I. ICPC Inference

Time limit: 3 seconds
Memory limit: 1024 megabytes

Let's consider an ICPC competition with rules almost identical to AMPPZ. Teams submit solutions, which are evaluated as correct or incorrect. The first correct submission for a problem increases the team's time penalty by the number of minutes[4] from the start of the competition plus **20** minutes for each previous incorrect submission[5] for this task. All submissions to a problem after the first correct one are ignored and do not affect the team's score. Teams are ranked in descending order by the number of correctly solved problems. Ties are resolved in ascending order of the total time penalty, and further ties are resolved randomly.

The competition has just ended! It lasted for L minutes and D teams submitted a total of N submissions for **13** problems prepared by the organizers. The top three teams with a positive score will stand on the podium. The jury decided not to spoil the surprise and did not look at the results of the submissions or even which problems the submissions were for. For each submission, they only knew the time and the team that submitted it. The jury is now wondering what the podium could look like, i.e., the sequence of the top three teams in the ranking, each of which has solved at least one problem.

Find the number of such possible podiums from the jury's perspective. We ignore cases where fewer than three teams have solved any problems.

Input

The first line of the input contains three integers: N, D and L ($3 \le N, D, L \le 200\,000$). These represent the number of submissions, the number of teams, and the duration of the competition in minutes, respectively.

In the following N lines, there are two integers in each line: d_i and t_i ($1 \le d_i \le D$, $1 \le t_i \le L$, $t_i \le t_{i+1}$). These integers denote the team index and the time of the i-th submission from the start of the competition. These submissions are sorted by time.

Output

Output a single integer – the number of possible ordered sets of three teams on the podium.

Examples

standard input	standard output
4 3 300 1 10 2 25 2 30 3 50	3
4 6 200000 6 1 6 1 1 2 2 2	4

Note

In the first sample test case, there are 3 possible podium configurations:

- $(1, 2, 3)$ – if each of the teams correctly solved one problem with total time penalties of $(10, 25, 50)$

[4] AMPPZ measures the submission time precise to seconds.
[5] In AMPPZ, only submissions that compile successfully increase the time penalty.

or $(10, 30, 50)$ or $(10, 50, 50)$. The last option assumes that team 2 submitted incorrect and then correct solutions for the same problem, resulting in a total time penalty of $30 + 20 = 50$.

- $(1, 3, 2)$ – again, with total time penalties of $(10, 50, 50)$, with the tiebreaker decided randomly in favor of team 3.
- $(2, 1, 3)$ – if team 2 correctly solved two problems, and the other teams solved one each.

In the second sample test case, the possible podium configurations are $(1, 2, 6)$, $(2, 1, 6)$, $(6, 1, 2)$, and $(6, 2, 1)$.

Problem J. Juliet Unifies Ones

Time limit: 2 seconds
Memory limit: 1024 megabytes

We call a binary string (consisting of ones and zeros) *unified* if all the ones form a contiguous (possibly empty) interval without any zeros in between. Examples of such strings are `0011110`, `1`, and `0000`. However, the binary strings `101` and `00100011` are not unified.

Juliet has a binary string S, and she is willing to remove some characters to make the string unified. When Juliet removes a character, the remaining characters slide to fill the gap.

How many characters must be removed from S to make the remaining characters form a unified binary string?

Input

The input consists of a single line containing the string S ($1 \leq |S| \leq 50$, $S_i =$ '`0`' or $S_i =$ '`1`').

Output

Output a single integer – the minimum possible number of removed characters.

Example

standard input	standard output
00011011001	2

Note

In the string `00011`<u>`0`</u>`11001`...

In the string $\texttt{00011}\underline{\texttt{0}}\texttt{1100}\underline{\texttt{1}}$, Juliet can remove two underlined characters to obtain the unified string `000111100`.

Problem K. Routing K-Codes

Time limit: 1 second
Memory limit: 1024 megabytes

In the internal network of the Bajtex company, there are n routers numbered from 1 to n. Some pairs of routers are connected by bidirectional links. Each link connects two different routers, and there is at most one direct link between any two routers. Every two routers are connected by a sequence of links in at least one way.

The system administrator is deploying a new algorithm for packet routing. This requires assigning a unique 32-bit routing code $K(a)$ to each router a ($0 \le K(a) < 2^{32}$, $K(a) \neq K(b)$ for $a \neq b$). If two routers are connected by a direct link, one of these codes should be half of the other, rounded down. In other words, if routers a and b are connected by a link then:

$$K(a) = \left\lfloor \frac{K(b)}{2} \right\rfloor \qquad \text{or} \qquad K(b) = \left\lfloor \frac{K(a)}{2} \right\rfloor$$

Check if it is possible to assign routing codes according to these requirements. If it is possible, calculate the minimum possible sum of these codes, $\sum_{i=1}^{n} K(i)$.

Input

The first line of the input contains two integers, n and m ($1 \le n \le 200\,000$, $n - 1 \le m \le 200\,000$), representing the number of routers and the number of links.

The next m lines describe the links. In the i-th of these lines there are two integers, a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \neq b_i$), describing a link between routers a_i and b_i. Each link appears in the input at most once (if $i \neq j$ then $(a_i, b_i) \neq (a_j, b_j)$ and $(a_i, b_i) \neq (b_j, a_j)$).

Output

If it is possible to assign unique routing codes correctly, the output should contain a single integer: the minimum possible sum of the codes $\sum_{i=1}^{n} K(i)$. If it is not possible, the output should consist of only the word `NIE`.

Examples

standard input	standard output
4 3 1 2 1 3 1 4	6
4 6 1 2 2 3 3 4 4 1 1 3 2 4	NIE

Note

In the first example, the routers can be assigned codes respectively: 1, 0, 2, 3, which sum up to 6.

In the second example it is not possible to assign codes.

Problem L. Random Numbers

Time limit: 2 seconds
Memory limit: 1024 megabytes

A **random** permutation of numbers from 1 to n is given. In other words, each number from 1 to n appears exactly once, and their order is random.

We are looking for *interesting* intervals, which are those where the sum of elements in the interval is equal to the square of its length. Formally, in the sequence $a_1, a_2, \ldots, a_n$, an interesting interval corresponds to the index range $[p, q]$ $(1 \le p \le q \le n)$ such that:

$$\left(\sum_{i=p}^{q} a_i\right) = (q - p + 1)^2$$

Count the number of interesting intervals.

Input

The first line of the input contains an integer t $(1 \le t \le 200,000)$, which represents the number of test cases. Each test case is described in two lines.

The first line of each test case contains an integer n $(1 \le n \le 200,000)$, which represents the length of the sequence.

The second line of each test case contains n different integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n,\ a_i \ne a_j$ for $i \ne j)$. The sequence is randomly selected, meaning each of the $n!$ sequences has an equal probability of being chosen, independently for different test cases. However, the organizers can choose the number t and the numbers n arbitrarily in each test case.

The sum of n over all test cases does not exceed $200,000$.

Output

The output should consist of t lines. The i-th line should contain a single integer – the number of interesting intervals in the i-th test case.

Example

standard input	standard output
2 3 2 1 3 5 3 4 2 5 1	2 2

Note

In the first test case, the interesting intervals are $[2, 2]$ (because $1 = 1^2$) and $[2, 3]$ (because $1 + 3 = 2^2$).

In the second test case, the interesting intervals are $[1, 3]$ (because $3 + 4 + 2 = 3^2$) and $[5, 5]$ (because $1 = 1^2$).

Problem M. Mathematics Championships

Time limit:	1 second
Memory limit:	1024 megabytes

In the Mathematical Championships, there are 2^n mathematicians, each with an initial fame value a_i, possibly negative. When a *Mathematical Duel* takes place between two mathematicians, the organizers determine the winner in a way known only to them. The fame of the loser remains unchanged, but the fame of the winner of the duel increases by the fame of the loser (`a[i] += a[j]`). Note that this can even lead to a decrease in the winner's fame!

The championship consists of n stages. In each stage, the organizers pair the participants. Within each pair, a Mathematical Duel takes place, and the winner advances to the next stage, while the loser is eliminated. After the first stage, there will be 2^{n-1} participants left, after the second stage, there will be 2^{n-2} participants left, and so on. Finally, after the nth stage, only one participant remains, and they are awarded a symbolic chocolate.

After the championship, an interview is planned with one of the participants, not necessarily the chocolate holder. The best candidate for the interview is the one among the 2^n mathematicians with the highest final fame. The prestige of the championship and the possibility of attracting sponsors for next year's edition depend on this interview. Help the organizers match the participants in each stage and select the winners to maximize the final fame of one of the mathematicians. Find this maximum value.

Input

The first line of the input contains a single integer n ($1 \le n \le 16$).

The second line contains 2^n integers $a_1, a_2, \ldots, a_{2^n}$. The i-th of these numbers is the initial fame of the i-th participant ($-10^6 \le a_i \le 10^6$).

Output

The output should contain a single integer – the maximum possible fame of one of the 2^n mathematicians after the end of the Championship.

Example

standard input	standard output
2 5 -1 2 -10	7

Note

In the first stage, the organizers create the following pairs of mathematicians (which is not necessarily the only optimal choice):

- In the pair with fame ($a_1 = 5, a_3 = 2$), let mathematician 3 win; his fame changes from 2 to $2+5 = 7$. Participant 1 is eliminated with a fame of 5.
- In the pair ($a_2 = -1, a_4 = -10$), let mathematician 2 win; his fame changes from -1 to $-1 + (-10) = -11$. Participant 4 is eliminated with a fame of -10.

In this scenario, the mathematicians with fame of 7 and -11 advance to the second stage. The organizers select the latter as the winner; their fame changes from -11 to $-11 + 7 = -4$. Participant 3 is eliminated with a fame of 7, and participant 2 receives the chocolate and has a final fame of -4.

The final fame of the four mathematicians are $5, -4, 7, -10$. The highest fame is 7, and this is the maximum possible value.