2024 KAIST RUN Spring Contest

Sponsored By:

Rules

- This contest is 2024 KAIST RUN Spring Contest.
- This contest is sponsored by Jane Street, Startlink, and Samsung Software Membership.
- You can only participate individually.
- Use of the network is prohibited during the competition, except for submitting source codes and accessing language reference sites, and using translation. Here are examples of allowed reference and translation sites:
  - `C/C++           : https://en.cppreference.com/w/`
  - `Java            : https://docs.oracle.com/javase/8/docs/api/`
  - `Python          : https://docs.python.org/`
  - `Kotlin          : https://kotlinlang.org/docs/reference/`
  - `Naver Papago    : https://papago.naver.com/`
  - `Google Translate: https://translate.google.com/`
- Use of pre-written code or template, any library is prohibited. You should only submit code written by yourself during contest time.
- The contest lasts 4 hours.
- The contest consists of 8 problems.
- Problems are ordered roughly by the difficulty of getting full points. You are strongly encouraged to read all the problems.
- Each problem consists of 1 or more subtasks, and each subtask is worth a certain number of points. Subtasks are easier versions of the problems with more restrictive constraints.
- Each subtask consists of several tests, and points for the subtask are awarded if every test is passed. The total number of points for each problem is 100 points.
- Each problem has time limit and memory limit. This means your problem should run in the given time and memory limit for each test.
- Every problem is guaranteed to be solvable using `C++17`. This is not guaranteed for any other language.
- The memory limit for every problem is 1024 MiB.
- Each participant's ranking is determined in the following way:
  - Penalty time = (Duration from contest start to the last submission that increased points)
  - Ranking = (# of participants with higher points) + (# of participants with same points and lower penalty time) + 1

Problem list

#	Problem name	Time limit (All languages)
A	RUN Number	1 second
B	Sequence and Queries	1 second
C	Construct a Graph	1 second
D	Closet	5 seconds
E	Two Histograms	2 seconds
F	Discount Event	4 seconds
G	One, Two, Three	1 second
H	Tree Kadane	3 seconds

Problem A. RUN Number

Time limit: 1 second

We call a positive integer x a **RUN** number if all digits of x are the same. For example, 4, 111, 888 888 are **RUN** numbers, while 27, 334, 100 000 are not.

Given an N-digit number K, represent K as a sum of at most $(N+1)$ **RUN** numbers. We can prove that the solution always exists.

Input

The first line contains an integer T — the number of test cases.

Each test case consists of one line containing two integers N and K separated by a space.

Output

For each test case,

- On the first line, print an integer M — the number of **RUN** numbers which add up to K. $(1 \le M \le N+1)$
- On the second line, print the M **RUN** numbers separated by a space. If there are multiple solutions, you may print any of them.

Constraints

- $1 \le T \le 100$
- $1 \le N \le 17$
- $10^{N-1} \le K < 10^N$. In other words, K is an N-digit number.

Subtasks

- Subtask 1 (5 points): $N \le 2$
- Subtask 2 (10 points): $N \le 5$
- Subtask 3 (25 points): Let $K = \overline{s_1 s_2 \ldots s_N}$. Then, $s_1 \le s_2 \le \cdots \le s_{N-1} \le s_N$. In other words, the digits of K are in non-decreasing order.
- Subtask 4 (60 points): No additional constraints.

Examples

standard input	standard output
2 4 2024 3 506	4 999 999 22 4 3 444 55 7
2 17 99999999999999999 9 333666999	1 99999999999999999 3 333333333 333333 333

Problem B. Sequence and Queries

Time limit: 1 second

You are given a sequence $(s_1, s_2, \ldots, s_n)$ of length n. The function f is defined as follows:

$$f(i,j,k) = \begin{cases} 1 & \text{if } s_{i+t} \leq s_{j+t} \text{ for all } 0 \leq t < k \\ 0 & \text{otherwise} \end{cases}$$

Print the value of $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{\min(n-i+1,n-j+1)} f(i,j,k)$.

Input

The first line contains an integer n — the length of the sequence.

The second line contains n integers $s_1, s_2, \ldots, s_n$ separated by a space.

Output

Print the value of $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{\min(n-i+1,n-j+1)} f(i,j,k)$.

Constraints

- $1 \leq n \leq 5\,000$
- $1 \leq s_x \leq 10^9$ $(1 \leq x \leq n)$

Subtasks

- Subtask 1 (8 points): $1 \leq n \leq 500$
- Subtask 2 (92 points): No additional constraints.

Examples

standard input	standard output
2 2 3	4
5 2 4 2 2 1	35

Note

The explanation for the first example is as follows:

- $f(1,1,1) = 1$
- $f(1,1,2) = 1$
- $f(1,2,1) = 1$
- $f(2,1,1) = 0$
- $f(2,2,1) = 1$

Problem C. Construct a Graph

Time limit: 1 second

You are given an $N \times N$ square matrix D. Your task is to construct an undirected connected graph with N vertices that fulfills the following conditions. Each vertex is numbered from 1 to N, and you may assign positive integer weights to each edge as desired.

- For all pairs of vertices (u, v), the length of the shortest path between u and v must be equal to $D_{u,v}$.
- The total sum of edge weights must be minimized.

Determine whether such a graph exists, and if so, print any graph that satisfies these conditions.

Input

The first line contains an integer N — the number of vertices.

The i-th of the next N lines contains N integers $D_{i,1}, D_{i,2}, \ldots, D_{i,N}$ separated by a space.

Output

If no graph satisfies the conditions, print -1.

If there exists a graph that satisfies the conditions:

- On the first line, print an integer M — the number of edges.
- On the i-th of the next M lines, print three integers u_i, v_i, c_i, separated by a space. These integers mean that the i-th edge connects two vertices u_i and v_i and its weight is c_i.
- There should be at most one edge for each pair of vertices, and the weight of each edge should not exceed 10^9.

Constraints

- $2 \leq N \leq 300$
- $D_{i,i} = 0$ $(1 \leq i \leq N)$
- $1 \leq D_{i,j} = D_{j,i} \leq 10^9$ $(1 \leq i < j \leq N)$
- $1 \leq u_i, v_i \leq N$ $(1 \leq i \leq M)$
- $u_i \neq v_i$ $(1 \leq i \leq M)$
- $1 \leq c_i \leq 10^9$ $(1 \leq i \leq M)$
- There should be at most one edge connecting a pair of vertices.

Subtasks

- Subtask 1 (9 points): In this subtask, you only need to determine whether there exists a graph satisfying the conditions. In other words, the checker only checks whether the output is -1 or not.
- Subtask 2 (19 points): $N \leq 50$
- Subtask 3 (72 points): No additional constraints.

Examples

standard input	standard output
3 0 1 2 1 0 3 2 3 0	2 1 2 1 1 3 2
3 0 1 3 1 0 1 3 1 0	-1

Problem D. Closet

Time limit: 5 seconds

Taein has a closet with N clothes hanging in it. Currently, the i-th clothes from the left has color c_i.

Taein considers his closet beautiful if there exists an integer k ($1 \le k \le N$) such that $c_1 \le c_2 \le \cdots \le c_k \ge c_{k+1} \ge \cdots \ge c_N$.

However, organizing the closet into a beautiful state is quite tedious. So, Taein decided to remove at most M clothes from the closet to make the remaining clothes **almost beautiful**.

After removing at most M clothes, let L be the number of remaining clothes, and let d_j be the color of the j-th clothes from the left. Taein decided to recognize two adjacent clothes as the same color if the difference in color between them is at most x. In other words, the closet is called **almost beautiful** if there exists an integer k ($1 \le k \le L$) for which the following holds:

- $d_j - d_{j+1} \le x$ for each integer j such that $1 \le j < k$.
- $d_{j+1} - d_j \le x$ for each integer j such that $k \le j < L$.

Let's find the smallest value of non-negative integer x that can make the closet **almost beautiful** by removing M or fewer clothes.

Input

The first line contains two integers N and M separated by a space.

The second line contains N integers $c_1, c_2, \ldots, c_N$ separated by a space.

Output

Print the smallest value of x ($x \ge 0$) that can make the closet **almost beautiful**.

Constraints

- $1 \le N \le 10^5$
- $0 \le M \le N$
- $1 \le c_i \le 10^9$ ($1 \le i \le N$)

Subtasks

- Subtask 1 (4 points): $M = 0$
- Subtask 2 (21 points): $1 \le N \le 1\,000$
- Subtask 3 (16 points): $1 \le c_i \le 100$ ($1 \le i \le N$)
- Subtask 4 (59 points): No additional constraints.

Example

standard input	standard output
10 2 4 2 7 15 3 11 12 10 2 6	2

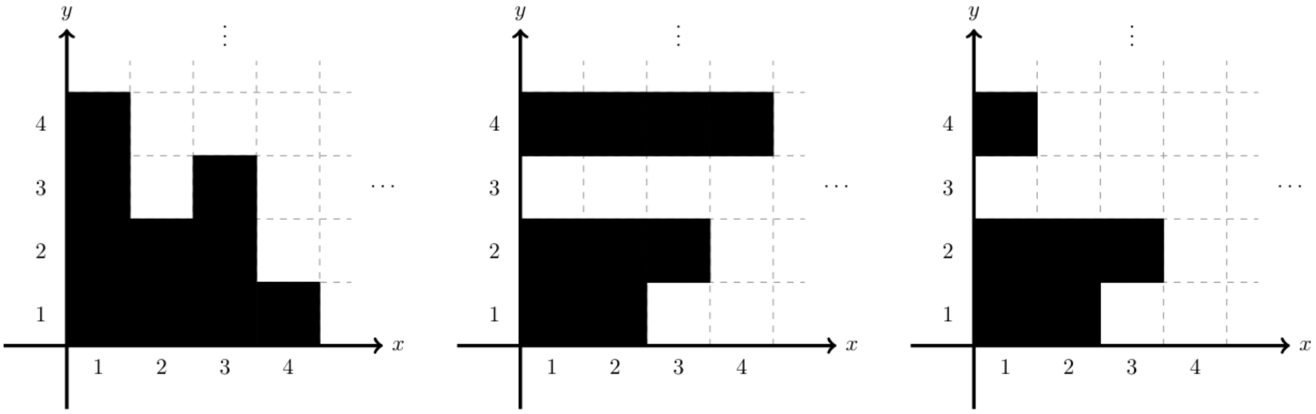
Note

When $x = 2$, Taein can make the closet **almost beautiful** by removing the 5th and 9th clothes from the left. There are no smaller values of x which make it possible to make the closet **almost beautiful**.

Problem E. Two Histograms

Time limit: 2 seconds

You are given three square grids with size $10^6 \times 10^6$. Each cell is labeled with x and y coordinates. The x coordinates are numbered from 1 to 10^6, left to right, and the y coordinates are numbered from 1 to 10^6, bottom to top. You must color each cell black or white.



An example of coloring the cells of three grids.

The first grid must have the shape of a histogram rising from the bottom. In other words, if a grid cell is colored black, the cell below must also be colored black.

The second grid must have the shape of a histogram progressing from left to right. In other words, if a grid cell is colored black, the cell left to it must also be colored black.

The third grid is colored using the first two grids. If a cell (x, y) is colored black in both of the first two grids, then cell (x, y) in the third grid is also colored black. Otherwise, the cell is colored white. This third grid becomes your final painting.

The painting is judged by N judges. Each judge will use a specific rectangular area of size $K \times 1$ for evaluation. The rectangular area used by the i-th judge is $[x_i, x_i + K - 1] \times [y_i, y_i]$. **The rectangular areas do not overlap each other.**

The i-th judge will reject the painting if cells (x_i, y_i) and $(x_i + K - 1, y_i)$ are the same color. If the colors are different, the judge will approve the painting. The judge will award a_i points if cell (x_i, y_i) is white and b_i points if it is black.

The painting must be approved by all judges to pass the evaluation. The painting's score is the sum of points awarded by all judges. Find the maximum possible score that can be achieved, considering all possible paintings that can pass the evaluation.

Input

The first line contains two integers N and K separated by a space.

The i-th of the next N lines contains four integers x_i, y_i, a_i, b_i separated by a space.

Output

If there is no possible painting that can pass the evaluation, print -1.

Otherwise, print the maximum score that your painting can get.

Constraints

- $1 \leq N \leq 3 \times 10^5$
- $2 \leq K \leq 10^6$

- $1 \le x_i \le 10^6 - K + 1$ $(1 \le i \le N)$
- $1 \le y_i \le 10^6$ $(1 \le i \le N)$
- $1 \le a_i, b_i \le 10^9$ $(1 \le i \le N)$
- The N rectangular areas $[x_i, x_i + K - 1] \times [y_i, y_i]$ $(1 \le i \le N)$ do not overlap.

Subtasks

- Subtask 1 (30 points): $K = 2$; $N \le 5\,000$
- Subtask 2 (30 points): $K = 2$
- Subtask 3 (40 points): No additional constraints.

Examples

standard input	standard output
5 2 1 1 1 2 3 1 1 10 5 1 5 6 2 2 3 2 4 2 5 9	26
6 3 1 1 2 4 5 1 4 9 2 3 7 4 5 3 3 1 1 5 5 7 4 5 6 4	36
10 2 7 2 2 4 4 4 6 3 1 5 1 4 3 5 2 8 5 2 4 3 6 4 4 2 1 2 1 4 5 6 9 7 7 1 6 3 4 3 8 7	51
10 3 4 2 5 2 10 2 8 10 1 2 1 4 12 1 8 6 6 3 7 10 8 1 1 9 11 3 5 5 7 2 10 5 3 3 6 4 4 1 9 4	72

Note

A rectangular area $[x_l, x_r] \times [y_l, y_r]$ refers to the area where $x_l \le x \le x_r$ and $y_l \le y \le y_r$.

Problem F. Discount Event

Time limit: 4 seconds

There is a country with N cities and $N-1$ roads. Each city is labeled with integers from 1 to N, and each road is labeled with integers from 1 to $N-1$. The i-th road connects cities A_i and B_i bidirectionally, and has travel costs W_i. You can travel along any pair of cities using the roads.

The distance between two cities is defined as the minimum cost of moving from one city to another.

As the president of a road company, you are planning a discount event for the holidays. There are a total of Q plans for the discount event. In the i-th plan, the discount is applied to all roads in the shortest path between cities X_i and Y_i, and the travel costs of the road become zero. For each discount event plan, compute the maximum value of the distance between any two cities.

Input

The first line contains an integer N — the number of cities.

The i-th of the next $N-1$ lines contains three integers A_i, B_i, and W_i separated by a space — the cities the i-th road connects and the road's travel cost.

The next line contains an integer Q — the number of plans.

The i-th of the next Q lines contains two integers X_i and Y_i separated by a space, describing the i-th plan.

Output

Print the answer along Q lines. In the i-th line, you should print the answer for the i-th query.

Constraints

- $2 \leq N \leq 100\,000$
- $1 \leq A_i, B_i \leq N$ $(1 \leq i \leq N-1)$
- $A_i \neq B_i$ $(1 \leq i \leq N-1)$
- $1 \leq W_i \leq 10^9$ $(1 \leq i \leq N-1)$
- It is possible to travel along any pair of cities using the roads.
- $1 \leq Q \leq 100\,000$
- $1 \leq X_i, Y_i \leq N$ $(1 \leq i \leq Q)$
- $X_i \neq Y_i$ $(1 \leq i \leq Q)$

Subtasks

- Subtask 1 (10 points): $N, Q \leq 2\,000$
- Subtask 2 (24 points): $X_i = 1$ $(1 \leq i \leq Q)$
- Subtask 3 (27 points): $Q = N-1$; $X_i = A_i$; $Y_i = B_i$ $(1 \leq i \leq N-1)$
- Subtask 4 (39 points): No additional constraints.

Examples

standard input	standard output
5 1 2 1 1 3 2 2 4 4 2 5 2 3 1 4 3 5 5 4	4 4 3
5 1 2 2 1 3 3 1 4 5 1 5 4 3 1 2 1 4 4 5	9 7 5
8 1 2 3 1 3 5 2 4 5 2 5 3 2 6 6 6 7 1 6 8 2 7 1 2 1 3 2 4 2 5 2 6 6 7 6 8	13 13 16 16 13 16 15
7 1 2 3 1 3 4 2 4 2 2 5 1 3 6 6 3 7 2 6 1 2 1 3 1 4 1 5 1 6 1 7	12 11 11 12 7 11

Problem G. One, Two, Three

Time limit: 1 second

You are given a sequence of N integers $a_1, a_2, \ldots, a_N$. Each element of a is 1, 2, or 3.

Each element of the sequence has a beauty, expressed as a positive integer. The i-th element has a beauty of b_i.

You can apply the following operation as many times as you want: choose a subsegment (contiguous elements) of the sequence and remove it if the sum of its elements is 4 or 8.

After applying the operations, we want to minimize the sum of elements. If there are multiple ways to minimize the sum, we want to maximize the sum of the beauty of the remaining elements.

Your task is to minimize the sum of remaining elements using an arbitrary number of operations. If there are many ways, you should maximize the sum of the beauty of the remaining elements. Please be aware that you don't have to print the sequence of operations for every test case. (Refer to the input and output specification for details.)

Input

Each test data contains one or more test cases. The first line contains an integer T — the number of test cases for this input file.

Each test case consists of four lines:

- The first line contains an integer N — the length of the original sequence.
- The second line contains a string $a_1 a_2 \ldots a_N$, representing the original sequence. Each letter of the string is either 1, 2, or 3.
- The third line contains N integers $b_1, b_2, \ldots, b_N$ separated by a space — the beauty of each element in the sequence.
- The fourth line contains an integer p — the type of each test case. For test cases such that $p = 1$, you only need to print the minimum sum of the remaining numbers and the maximum sum of beauty of the remaining numbers. For test cases such that $p = 2$, you need to print the sequence of operations additionally.

Output

For each test case, first print the minimum sum of the remaining numbers and the maximum sum of the beauty of the remaining numbers.

If $p = 2$, you should additionally print the sequence of operations. In the second line, print K, the number of operations.

In the following K lines, you should print the information about the operations. In the i-th line among them, first, print the number of removed elements in the i-th operation, and next, print their indices in increasing order. The indices should form a subsegment in the sequence before applying the i-th operation. Printed indices should be indices from the initial sequence.

Constraints

- $1 \le T \le 2\,000$
- $N \ge 1$ for each test case
- $\sum N \le 3 \times 10^5$
- $1 \le a_i \le 3$ $(1 \le i \le N)$ for each test case
- $1 \le b_i \le 99$ $(1 \le i \le N)$ for each test case
- $1 \le p \le 2$ for each test case

Subtasks

- Subtask 1 (20 points): $\sum N \le 3\,000$. $p = 1$ for each test case.

- Subtask 2 (30 points): $\sum N \leq 3\,000$
- Subtask 3 (20 points): $p = 1$ for each test case
- Subtask 4 (30 points): No additional constraints.

Example

standard input	standard output
6 3 121 1 2 3 2 3 221 4 5 6 2 3 123 7 8 9 1 6 123221 10 11 12 13 14 15 2 6 312322 15 14 13 12 11 10 2 6 213312 9 8 7 5 5 1 2	0 0 1 3 1 2 3 1 6 1 2 1 2 6 24 3 29 1 4 1 2 3 4 5 25 2 2 5 6 2 1 2 0 0 3 2 2 3 2 4 5 2 1 6

Problem H. Tree Kadane

Time limit: 3 seconds

You are given a tree T with N vertices numbered 1 to N. Each vertex i has an integer weight A_i.

A **connected subset** S of T is a nonempty subset of vertices in T such that, for any two vertices a, b in S, there exists a path between a, b in T that only uses the vertices from S.

You need to perform the following Q queries:

- k x : change A_k to x and print the maximum value of $\sum_{i \in S} A_i$ where S is a **connected subset** of T.

Input

The first line contains an integer N — the number of vertices.

The second line contains N integers $A_1, A_2, \ldots, A_N$ separated by a space — the weights of each vertex.

The i-th of the next $N-1$ lines contains two integers u_i and v_i separated by a space — the endpoints of the i-th edge.

The next line contains an integer Q — the number of queries.

Each of the next Q lines contains two integers k and x separated by a space — arguments of each query.

Output

Print Q lines. In each line, print the answer to each query.

Constraints

- $1 \le N \le 10^5$
- $-10^9 \le A_i \le 10^9$ $(1 \le i \le N)$
- $1 \le u_i, v_i \le N$ $(1 \le i \le N-1)$
- The given graph is a tree.
- $1 \le Q \le 10^5$
- $1 \le k \le N$ for each query
- $-10^9 \le x \le 10^9$ for each query

- Subtask 1 (10 points): $N, Q \le 10^3$
- Subtask 2 (20 points): Each vertex has a degree less than 3.
- Subtask 3 (70 points): No additional constraints.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5
3 -1 -4 -1 -5
1 2
1 3
3 4
3 5
5
1 -2
3 2
4 1
2 5
1 -9</pre></td><td><pre>-1
2
3
6
5</pre></td></tr>
</table>