2026 ACM-ICPC SCUT SCHOOL CONTEST

2026 ACM-ICPC SCUT SCHOOL CONTEST(Spring Season Tournament)

SCUT, 29 MARCH 2026

Judges and Problem Setters

- scutkk
- hrmm
- Shuohan_Huang
- hei_yu_bai
- Haotian_Lyu

Basic Matrix Recurrence Practice

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	8 megabytes

Define the column vector:

$$v_n = (a_n, b_n, c_n, d_n)^T$$

Initial vectors:

$$v_0 = (1, 2, 3, 4)^T, \quad v_1 = (5, 6, 7, 8)^T$$

For all $n \geq 1$:

$$v_{n+1} \equiv P v_n + Q\big(v_{n-1} \circ v_{n-1}\big) \pmod{61}$$

P, Q are 4×4 constant matrices. The operator $\circ$ denotes Hadamard (element-wise) product.

The following conventions are used throughout this statement:

$$x \circ y = (x_1 y_1, x_2 y_2, x_3 y_3, x_4 y_4)^T, \qquad (Mx)_r = \left(\sum_{c=1}^{4} M_{r,c} x_c\right) \bmod 61.$$

Hence,

$$v_{n-1} \circ v_{n-1} = (a_{n-1}^2, b_{n-1}^2, c_{n-1}^2, d_{n-1}^2)^T.$$

All additions and multiplications in the recurrence are performed modulo 61.

To avoid massive I/O overhead, queries are processed online and only one final checksum is printed.

Maintain a 64-bit unsigned integer `rnd_seed` (initialized by input `seed`) and generate random numbers by:

```
uint64_t next_rand() {
    rnd_seed ^= rnd_seed << 13;
    rnd_seed ^= rnd_seed >> 7;
    rnd_seed ^= rnd_seed << 17;
    return rnd_seed;
}
```

Before query 1, set `last_ans` $= 0$ and `final_hash` $= 0$. For the i-th query ($1 \leq i \leq q$), do the following:

1. Generate:
$$n_i = (\texttt{next\_rand()} \oplus \texttt{last\_ans}) \bmod 10^{18}$$
2. Compute $v_{n_i} = (a, b, c, d)^T$, then compress:
$$\texttt{current\_ans} = a \mid (b \ll 6) \mid (c \ll 12) \mid (d \ll 18)$$
3. Update:
$$\texttt{last\_ans} \leftarrow \texttt{current\_ans}$$
$$\texttt{final\_hash} \leftarrow \texttt{final\_hash} \oplus (\texttt{current\_ans} \cdot i)$$

Bit operations are standard unsigned-integer operations: $\oplus$ is bitwise XOR, $\ll$ is left shift, and $\mid$ is bitwise OR.

Input

The first 4 lines contain matrix P, each line has 4 integers in $[0, 60]$.

The next 4 lines contain matrix Q, each line has 4 integers in $[0, 60]$.

The last line contains two integers `q` and `seed`: $1 \leq q \leq 10^7$ and $0 \leq \texttt{seed} < 2^{64}$.

In official tests, elements of P and Q are generated independently and uniformly at random in $[0, 60]$.The number of official test files does not exceed 30.

Output

Print a single 64-bit unsigned integer, which is the final value of `final_hash` after all q queries are processed.

Examples

standard input	standard output
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 1 5 7	42019442
6 5 4 3 2 6 5 4 3 2 6 5 4 3 2 6 6 6 5 5 4 4 3 3 2 2 1 1 5 1 4 2 5 1844674407370955161	67023929

Note

Strict Memory Limit Warning: The memory limit for this problem is extremely tight.

The Light Boat Has Passed Ten Thousand Mountains

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The legendary competitive programming team **Chickémon GO!** from South China University of Technology (SCUT) has had a remarkable journey. Half a year ago, at the CCPC Jinan Regional Contest, they unfortunately left without a medal. However, they did not lose heart. After months of rigorous training, they finally won a Silver Medal at the ICPC ECFinal, proving their status as the top team of SCUT. As the poem says, *"The light boat has passed ten thousand mountains"* — they have overcome countless difficulties to reach this point.

To document this journey of rebirth, the team captain collected n historical performance indices from their training contests. Each index a_i represents the difficulty level they overcame in the i-th contest.

They want to set a **baseline score** x to summarize their current level. The baseline must satisfy the following conditions:

1. $1 \le x \le 10^9$
2. Exactly k elements of the given sequence are less than or equal to x (i.e., exactly k indices i satisfy $a_i \le x$).

Note that the sequence can contain equal elements.

Your task is to find such an integer x. If there are multiple valid answers, print **the minimum** of them. If no such integer exists, print `-1`.

Input

The first line contains a single integer T ($1 \le T \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$, $0 \le k \le n$) — the number of historical records and the required count of records less than or equal to the baseline.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the historical performance indices.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the answer on a separate line.

Print **the minimum** integer x in the range $[1, 10^9]$ such that exactly k elements of the given sequence are less than or equal to x. If there is no such x, print `-1`.

Example

standard input	standard output
3 7 4 8 1 2 2 3 10 11 5 0 1 3 5 7 9 6 6 4 2 7 1 9 3	3 -1 9

Leyline Resonance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a tree consisting of n vertices, rooted at vertex 1. Each vertex i has an initial value a_i. For each vertex i from 2 to n, there is a directed edge from its parent p_i to i with a weight w_i.

You need to assign a real number x_i to each vertex i to minimize the total cost. The total cost is defined as the sum of the adjustment costs for all vertices and the penalty costs for all edges:

$$Cost = \sum_{i=1}^{n} |x_i - a_i| + \sum_{i=2}^{n} w_i \cdot \max(0, x_{p_i} - x_i)$$

Print the minimum possible total cost. It can be proven that the minimum cost is always an integer.

Input

The first line contains a single integer T ($1 \le T \le 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of vertices in the tree.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the initial values of the vertices.

Then $n - 1$ lines follow. The i-th of these lines ($1 \le i \le n - 1$) contains two integers p_{i+1} and w_{i+1} ($1 \le p_{i+1} \le i$, $0 \le w_{i+1} \le 10^5$) — the parent of vertex $i + 1$ and the weight of the edge connecting them.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer on a new line — the minimum possible total cost.

Example

standard input	standard output
3 1 5 4 5 1 6 2 1 3 2 0 3 5 7 3 -5 8 1 -2 10 -7 1 4 1 4 2 1 2 9 3 0 3 6	0 8 23

SCUT Classroom Relocation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

South China University of Technology (SCUT) has m classrooms available for self-study. The j-th classroom currently has c_j spare seats.

There are n study groups on campus. The i-th study group consists of q_i students and is initially assigned to their primary classroom p_i.

Due to a sudden campus-wide air conditioning maintenance drill, **all** study groups must be evacuated from their primary classrooms. Specifically, each student from the i-th study group must be relocated to any classroom j such that $j \neq p_i$. Students from the same group can be split and sent to different classrooms. Multiple students from various groups can be assigned to the same classroom, provided that the total number of newly added students does not exceed the classroom's spare capacity c_j.

Given the capacities of the classrooms and the sizes of the study groups, determine if it is possible to successfully relocate all students without exceeding the spare capacity of any classroom.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10^6$) — the number of study groups and the number of classrooms, respectively.

The second line of each test case contains m integers $c_1, c_2, \ldots, c_m$ ($1 \leq c_j \leq 10^9$) — the spare capacities of the classrooms.

The following n lines describe the study groups. The i-th of these lines contains two integers q_i and p_i ($1 \leq q_i \leq 10^9$, $1 \leq p_i \leq m$) — the number of students in the i-th group and the index of their primary classroom.

It is guaranteed that the sum of n and the sum of m over all test cases do not exceed 10^6.

Output

For each test case, print the answer on a new line — `YES` if it is possible to successfully relocate all students, and `NO` otherwise.

Example

standard input	standard output
4 1 1 10 1 1 2 3 2 2 2 4 1 3 2 3 3 5 3 4 5 1 3 1 1 2 2 2 3 3 3 1 3 2	NO NO NO YES

Phantoms of the XOR Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given an array A of length n consisting of distinct non-negative integers.

Consider a complete undirected graph G with n vertices. The weight of an edge between vertices i and j ($1 \le i < j \le n$) is defined as $A_i \oplus A_j$, where $\oplus$ denotes the bitwise XOR operation.

Since there can be multiple Minimum Spanning Trees (MST) in graph G, we define a new graph G_{MST} which contains all the vertices of G, and its edge set consists of all edges that belong to **at least one** Minimum Spanning Tree of G.

Calculate and print the total number of edges in G_{MST}.

Input

The first line of the input contains a single integer T ($1 \le T \le 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array.

The second line of each test case contains n distinct integers $A_1, A_2, \ldots, A_n$ ($0 \le A_i < 2^{30}$) — the elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer on a new line — the total number of edges in the graph G_{MST}.

Example

standard input	standard output
4 1 0 2 1 2 4 0 1 2 3 6 0 2 5 7 8 10	0 1 4 7

PigeonG's Encoded Sequences

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

PigeonG is a **prodigious** PhD student at the School of Software Engineering, South China University of Technology, working on AI-for-medicine research. In one of his projects, biological fragments are transformed into symbolic **encoded sequences** for downstream computational analysis. He is currently investigating the **Linguistic Complexity** of fused encoded fragments.

PigeonG has a database of n encoded fragments, denoted as $S_1, S_2, \ldots, S_n$. Each fragment is represented as a string over $\{\texttt{A}, \texttt{B}, \ldots, \texttt{Z}\}$. He needs to evaluate the complexity of various fusion combinations. The complexity is defined as the number of **distinct substrings** in the fused sequence.

Due to the massive size of the dataset, his current Python script is too slow to handle the workload. **He asks for your help** to design an efficient algorithm to speed up the research.

Task: PigeonG will provide you with q queries. In each query, he gives you two indices id_1 and id_2. Your task is to calculate the number of distinct substrings in the concatenated string

$$T = S_{id_1} \circ S_{id_2}$$

where $\circ$ denotes string concatenation.

Definitions:

- **Substring:** a non-empty contiguous segment of a string.
- **Distinct:** two substrings are distinct if their string contents are different.

Input

- The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$).
- The next n lines each contain a string over $\{\texttt{A}, \texttt{B}, \ldots, \texttt{Z}\}$, denoting S_i, with $1 \leq |S_i| \leq 2 \times 10^5$.
- It is guaranteed that $\sum_{i=1}^{n} |S_i| \leq 2 \times 10^5$.
- The next line contains an integer q ($1 \leq q \leq 2 \times 10^5$).
- The next q lines each contain two integers id_1, id_2 ($1 \leq id_1, id_2 \leq n$).
- **It is guaranteed that each character of S_i is chosen uniformly at random from $\{\texttt{A}, \texttt{B}, \ldots, \texttt{Z}\}$.**

Output

For each query, output one integer on a separate line: the number of distinct substrings in the corresponding concatenated string.

Example

standard input	standard output
3 SCUT SCUTSKY PIGEONG 5 1 2 2 3 3 1 1 1 2 2	51 103 65 26 76

Delete or not

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a strictly increasing sequence of n positive integers $A = [a_1, a_2, \ldots, a_n]$.

You need to answer q independent queries. In each query, you are given an interval $[l, r]$ $(1 \le l \le r \le n)$. Let B be the contiguous subsegment $[a_l, a_{l+1}, \ldots, a_r]$.

You can perform the following operation on the sequence B any number of times (possibly zero):

- Choose an index i $(2 \le i \le |B| - 1)$ such that $2B_i \le B_{i-1} + B_{i+1}$.
- Remove B_i from the sequence B. After the removal, the remaining elements are concatenated without changing their relative order, and the length of B decreases by 1 (i.e., the original B_{i-1} and B_{i+1} become adjacent).

For each query, find the minimum possible length of the sequence B after performing any number of valid operations.

Input

The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases.

The first line of each test case contains two integers n and q $(1 \le n, q \le 3 \cdot 10^5)$ — the length of the sequence and the number of queries.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_1 < a_2 < \ldots < a_n \le 10^9)$ — the elements of the sequence.

Each of the next q lines contains two integers l and r $(1 \le l \le r \le n)$ — the boundaries of the query.

It is guaranteed that the sum of n and the sum of q over all test cases do not exceed $3 \cdot 10^5$.

Output

For each query, output a single integer on a new line — the minimum possible length of the sequence B after any number of valid operations.

Example

standard input	standard output
2 5 4 1 5 6 8 9 1 5 1 3 2 5 4 5 6 5 2 3 7 8 10 15 1 6 2 5 1 4 3 6 5 5	4 3 3 2 2 3 3 2 1

Substring Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Given a string s, two players **Hiraethsoul** and **hjh** play the following game:

- Initially, both players have empty strings a and b.
- The two players take turns, and **Hiraethsoul** moves first.
- In each move, the current player removes exactly one character from either the **front** or the **back** of string s.
- The removed character is appended to the end of the player's own string.
- The game ends when s becomes empty.

After the game ends:

- If string a contains the substring `"scut"`, then **Hiraethsoul** wins.
- Otherwise, **hjh** wins.

Determine the winner assuming both players play optimally.

Input

The first line contains a single integer q ($1 \le q \le 5 \times 10^4$), the number of test cases.

For each test case:

The first line contains a single integer n ($1 \le n \le 10^4$), the length of the string.

The second line contains a string s of length n, consisting only of lowercase letters `s`, `c`, `u`, and `t`.

It is guaranteed that the sum of all n over all test cases does not exceed 5×10^4.

Output

For each test case, print `Yes` if **Hiraethsoul** wins; otherwise print `No`.

Example

standard input	standard output
2 7 scuttuc 5 scuts	Yes No

Note

Players always append the chosen character to the end of their own string. The substring condition is checked only after the game ends.
A substring is defined as a contiguous sequence of characters within a string.

Parallel Pipeline Scheduling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Hiraethsoul is designing a parallel processing pipeline.

There are N sequential tasks, where the i-th task requires T_i units of time. There are K local processing cores and one cloud server available.

Since the tasks must be executed in order, the N tasks must be partitioned into M contiguous segments, where $1 \leq M \leq K$. Each segment $[L_j, R_j]$ is assigned to one core. The segments are pairwise disjoint and together cover the entire range $[1, N]$.

For each core:

- Exactly one task from its assigned segment must be offloaded to the cloud server.
- The local processing time of this core equals the total processing time of its segment minus the processing time of the offloaded task.

The cloud server can process all submitted tasks in parallel. Its total processing time is defined as the maximum processing time among all tasks offloaded to the cloud.

Formally, let:

- $Core_j$ denote the local processing time of the j-th core;
- $cloud$ denote the cloud processing time.

The total system processing time is defined as

$$\max\left(\max_{1\leq j\leq M} Core_j,\ cloud\right).$$

Hiraethsoul wants to determine a partition of the tasks and the offloaded task in each segment such that the total system processing time is minimized.

Input

The first line contains two integers N and K ($1 \leq K \leq N \leq 10^6$).

The second line contains N integers $T_1, T_2, \ldots, T_N$ ($1 \leq T_i \leq 10^9$).

Output

Print a single integer — the minimum possible total completion time.

Examples

standard input	standard output
10 8 46 41 49 46 23 12 46 49 29 12	49
10 3 491 372 450 40 446 73 232 439 420 483	822

GCD and LCM Subsequences

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given an array A of length N.

Each element satisfies $1 \leq A_i \leq 10^{12}$.

You are also given two integers G and L.

Your task is to count the number of non-empty subsequences of A such that:

- The greatest common divisor (GCD) of all elements in the subsequence is exactly G.
- The least common multiple (LCM) of all elements in the subsequence is exactly L.

Since the answer may be large, output it modulo 998244353.

Input

The first line contains a single integer N ($1 \leq N \leq 5 \times 10^5$).

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^{12}$).

The third line contains two integers G and L ($1 \leq G, L \leq 10^{12}$).

Output

Print a single integer — the number of valid subsequences modulo 998244353.

Examples

standard input	standard output
5 50 4 50 4 100 2 100	18
5 8 2 2 1 2 1 8	8

Note

A subsequence of an array is obtained by deleting zero or more elements without changing the relative order of the remaining elements. The subsequence must be non-empty.

The GCD of a sequence is the greatest common divisor of all its elements. The LCM of a sequence is the least common multiple of all its elements.

It is not guaranteed that G divides L. If no valid subsequence exists, print 0.

Choose

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

`scutkk` hasn't done any exercises for a long time. To test his current level, `heiyubai` decided to give him a problem.

`heiyubai` provided `scutkk` with two arrays a and b of length n. He required that for all $1 \leq i \leq n$, `scutkk` must choose exactly one number from a_i and b_i, such that the **XOR sum** of all the numbers he chooses is maximized.

Since `scutkk` can't solve it, he asks you to help him solve this problem.

Input

The first line contains a single integer n ($1 \leq n \leq 10^6$) - the length of the array.

The second line contains n integer $a_1, a_2, ..., a_n$ ($0 \leq a_i < 2^{63}$)

The third line contains n integer $b_1, b_2, ..., b_n$ ($0 \leq b_i < 2^{63}$)

Output

Print the **Maximum XOR sum** of the number that you choose between a_i and b_i for all $1 \leq i \leq n$

Example

standard input	standard output
2 2 3 1 1	3

Matrix Construction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

`heiyubai` gave `repeator` a problem.

He gave `repeator` a matrix of size $h \times d$ and an array a of length n.

Let the i-th element of the array be a_i, which means *repeator* has a_1 copies of 1, a_2 copies of 2, ..., a_n copies of n.

`heiyubai` requires `repeator` to fill the matrix with these numbers, with the requirement that **no two identical numbers can appear in the same row, and no two identical numbers can appear in the same column**.

If it is impossible, output -1. Otherwise, output the matrix. If there are multiple valid solutions, output any one of them.

Input

The first line contains two integers h and d ($1 \leq h, d \leq 10^6, 1 \leq h \times d \leq 10^6$).

The second line contains a single integer n ($1 \leq n \leq 10^6$)

The third line contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i < 2 \times 10^6$)

Output

If it is impossible, output -1. Otherwise, output the matrix. If there are multiple valid solutions, output any one of them.

Examples

standard input	standard output
2 4 4 2 2 2 2	1 3 4 2 3 1 2 4
2 4 4 2 2 2 1	-1