The 2026 ICPC Asia Topi On-line Preliminary Programming Contest

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- Do not create disturbance or move around unnecessarily in the arena.
- If you have any question regarding the problems, send a clarification from the judges using DOMJudge.
- There would be no internet access and mobile phones are also not allowed.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of the main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file names should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with the judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- Your programs must meet the time constraints given at each problem.
- The decision of judges will be absolutely final.
- ***Plagiarism or use of Generative AI tools is strictly prohibited and will result in immediate disqualification.***

Problem A
Bit Shuffle

Time limit: 1 second

A **permutation** π of $\{0, 1, \ldots, k-1\}$ **encodes** a k-bit integer a as follows: bit i of a is moved to bit position $\pi(i)$, for every i. Bit 0 is the least significant bit (lsb). Formally, the encoding of a is the integer b whose bit $\pi(i)$ equals bit i of a for all $0 \le i < k$. Treat both a and b as exactly k bits, including leading zero bits.

You are given n pairs of integers $(a_1, b_1), \ldots, (a_n, b_n)$. We want one fixed permutation π under which b_j is the encoding of a_j for every j. If several permutations qualify, the answer is the lexicographically smallest one: the one with the smallest $\pi(0)$, then the smallest $\pi(1)$, and so on.

Task: Output the lexicographically smallest such permutation π, or -1 if no permutation encodes every a_j as b_j.

Example:

Let $k = 3$ and the pairs be $(3, 6)$ and $(4, 1)$. The numbers in binary, listed from bit 0 to bit 2:

Number	Value	Bit 0	Bit 1	Bit 2
a_1	3	1	1	0
b_1	6	0	1	1
a_2	4	0	0	1
b_2	1	1	0	0

It can be shown that the lexicographically smallest permutation π that encodes each a_j to b_j is: $1\ 2\ 0$. That is, $\pi(0) = 1, \pi(1) = 2, \pi(2) = 0$.

1. Bit 0 of $a_1 = 1$ moves to bit $\pi(0) = 1$ of b_1.
   Bit 0 of $a_2 = 0$ moves to bit $\pi(0) = 1$ of b_2.
2. Bit 1 of $a_1 = 1$ moves to bit $\pi(1) = 2$ of b_1.
   Bit 1 of $a_2 = 0$ moves to bit $\pi(1) = 2$ of b_2.
3. Bit 2 of $a_1 = 0$ moves to bit $\pi(2) = 0$ of b_1.
   Bit 2 of $a_2 = 1$ moves to bit $\pi(2) = 0$ of b_2.

The permutation $2\ 1\ 0$ is also valid, but it is lexicographically larger.

Input

First line: two integers n and k ($1 \le n \le 2 \cdot 10^4, 1 \le k \le 60$).

Next n lines: two integers a_j and b_j ($0 \le a_j, b_j < 2^k$) on the j-th line. The same integer or pair may appear more than once.

Output

Print k space-separated integers $\pi(0), \pi(1), \ldots, \pi(k-1)$: the lexicographically smallest valid permutation. If there is none, print -1.

Sample Input 1	Sample Output 1
2 3 3 6 4 1	1 2 0

Sample Input 2	Sample Output 2
1 2 1 3	-1

Sample Input 3	Sample Output 3
3 3 1 2 2 4 4 1	1 2 0

Sample Input 4	Sample Output 4
3 4 6 12 9 10 6 12	-1

Problem B
Beacon Maintenance Network

Time limit: 3 seconds

A monitoring pipeline has n beacons in a line, numbered 1 through n. Each beacon is either **active** or **inactive**.

The engineer can service any active beacon. Servicing the beacon at position i does two things:

- Removes that beacon permanently, leaving a gap.
- Toggles each beacon at positions $i - 1$ and $i + 1$ that is still present. An active beacon becomes inactive, and an inactive beacon becomes active.

Inactive beacons cannot be serviced. Beacons keep their original positions: gaps never close, and an operation does not affect beacons across a gap.

The remaining beacons form **segments**. A segment is a maximal group of consecutive positions where no beacon has been removed. The pipeline is fully shut down when all beacons have been removed.

The pipeline also has an automated recovery system that can perform the same service operations. The engineer wants to reach a state where **every remaining segment contains exactly one active beacon**. From that point, later service operations cannot prevent a full shutdown.

Only the engineer acts during the operations used to reach this state. **The recovery system does not act during this sequence.**

The initial state is given by a binary string of length n, where 1 means active and 0 means inactive. Initially, no beacons have been removed.

An initial state is called **serviceable with budget f** if the engineer can reach the required state using **at most f operations**. Using zero operations is allowed. If no beacons remain, the required condition is also satisfied.

For each query (n, f), count the binary strings of length n that are serviceable with budget f.

Facts you may use

You may use the following facts without proving them:

- For an initially unbroken pipeline, **there exists an order of legal operations that removes all beacons if and only if the number of active beacons is odd**. This does not mean that every legal order succeeds.
- A segment with an even number of active beacons, including zero, can never be fully removed. Any service operation within such a segment leaves at least one segment with an even number of active beacons.
- If a segment contains more than one active beacon, the recovery system can choose a single service operation that leaves a segment with an even number of active beacons. Full shutdown then becomes impossible.
- If every remaining segment contains exactly one active beacon, any legal service operation leaves the same condition true for all remaining segments. Therefore, continuing to service active beacons will remove the entire pipeline, regardless of which active beacon is chosen at each step.

Input

The first line contains an integer q ($1 \le q \le 100000$), the number of queries.

Each of the next q lines contains two integers n and f ($1 \le n \le 1000000$, $0 \le f \le n$): the pipeline length and the maximum number of operations the engineer may use.

For each query, define

$$j_{max} = \lfloor (n - 1) / 2 \rfloor, f_c = \min(f, j_{max}),$$

and

$$w = \min(f_c + 1, j_{max} - f_c + 1).$$

Queries may repeat. Across all distinct pairs (n, f_c) in one input file, the sum of w is at most $5 \cdot 10^7$. Count each distinct pair only once when calculating this sum.

Output

Print q integers in query order.

For each query, print the number of serviceable binary strings of length n, modulo 1000000007.

Sample Input 1	Sample Output 1
4 3 0 3 1 3 2 3 3	3 4 4 4

Sample Input 2	Sample Output 2
5 4 0 4 1 4 2 4 3 4 4	4 8 8 8 8

Problem C
Integer After Removing Similar Binaries

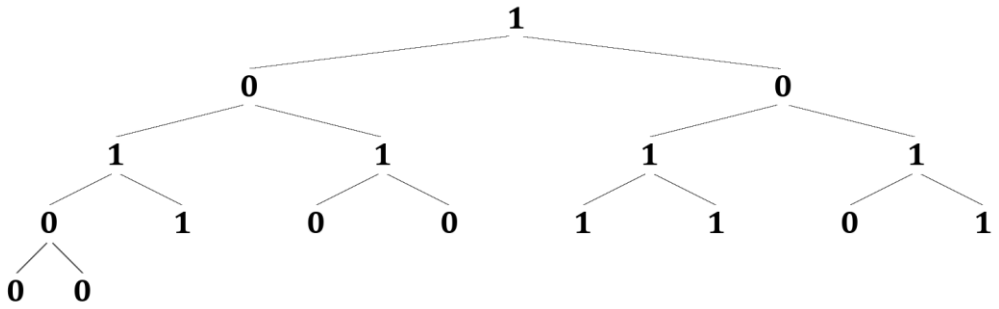
Time limit: 1 second

You are given a positive integer $\boldsymbol{n}$. Write it in binary, most significant bit first, without leading zeros. Place its bits in a complete binary tree, filling each level from left to right.

A subtree consists of a node and all its descendants. Remove an entire subtree, including its root, if it contains at least three nodes and all its bits are the same.

Visit the tree in preorder: root, left subtree, then right subtree. Check each node before its children. If its subtree is removed, skip all its nodes. Otherwise, keep its root and continue. Do not check a node again after pruning its descendants.

The diagram shows the tree for $\boldsymbol{n = 81204}$.



Removing the three-node all-zero subtree on the left and the three-node all-one subtree on the right leaves the preorder sequence $\boldsymbol{10111000101_2 = 1477}$.

Input

A single positive integer $\boldsymbol{n}$ $\boldsymbol{(1 \leq n \leq 10^9)}$

Output

Print the decimal value of the remaining preorder bit sequence. If no nodes remain, print $\boldsymbol{0}$.

Sample Input 1	Sample Output 1
81204	1477
Sample Input 2	**Sample Output 2**
128	1
Sample Input 3	**Sample Output 3**
1028	17
Sample Input 4	**Sample Output 4**
1031	139

Problem D
Mulk Ke Wasee Tar Mafad Mein

Time limit: 3 seconds

The provincial government has been suspended. The public has been assured that its mandate remains respected. It is merely being respected from a different office.

The outgoing administration must now vacate the premises. Statements have been issued, objections have been recorded, and someone has finally asked the practical question:

"Gaariyon ka intezam kis ne karna tha?"

Unfortunately, that question was addressed to you.

There are n outgoing babus waiting in a line. Babu i carries a_i units of luggage. Their order must be preserved. After all, protocol is the one arrangement that can never be suspended.

Normally, the roads would be lined with containers, and some babus might find themselves on a different journey altogether. For this problem, assume the roads are clear.

Divide the line into nonempty consecutive groups, assigning each group to one hired vehicle. Every babu must travel exactly once, with all their luggage. There are limits to austerity.

White Corollas and black Vigos are available in any quantity. A white Corolla has luggage capacity C_c and costs P_c to hire. A black Vigo has luggage capacity C_v and costs P_v. Either type may be larger or cheaper. The procurement committee would have preferred to know this before approving the bill. The briefcase under the table settled the matter.

Every group's total luggage must fit within its vehicle's capacity.

The new administration has also announced an austerity policy: each vehicle must leave with at least half its luggage capacity in use. At most one vehicle in the entire convoy may carry less than half its capacity. This exception may occur anywhere in the convoy. Exactly half-full is acceptable.

The policy applies equally to everyone. The exception has been approved separately. Do not let any babu tell you otherwise.

Find the minimum total hiring cost. If no valid arrangement exists, print -1.

You must work with the input provided. You may not suspend the constraints and appoint a more cooperative test case. Above your pay grade.

Input

The first line contains one integer n ($1 \le n \le 2000$).

The second line contains four integers C_c, P_c, C_v and P_v ($1 \le C_c, P_c, C_v, P_v \le 10^9$), in that order.

The third line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

Print one integer: the minimum total hiring cost, or -1 if no valid transportation is possible.

Sample Input 1	Sample Output 1
5 3 2 8 3 3 3 8 2 3	9

Sample Input 2	Sample Output 2
3 4 10 4 10 4 4 1	30

Sample Input 3	Sample Output 3
4 4 10 4 10 4 1 4 1	-1

Problem E
Mergers & Acquisitions

Time limit: 1 second

There are N corporations arranged in a line in Silicon Valley, numbered from 1 to N from left to right. Corporation i has a bank balance of P_i billion.

Because humans are inherently greedy, the CEO of a corporation may take over the corporation immediately to its left if and only if its bank balance is strictly greater than that corporation's balance.

Suppose corporation B takes over corporation A. Corporation A is permanently dissolved. Corporation B takes over A's territory, bringing B adjacent to whichever corporation was immediately to the left of A.

Corporation B's bank balance stays unchanged: it does not receive A's balance. It can then take over its new left neighbour if its balance is strictly greater. Corporations with equal balances cannot take over each other.

You are given Q independent queries. Each query contains two integers L and R. For that query, consider only the original corporations from position L to position R, inclusive. Corporations outside this segment do not participate.

The corporations perform takeovers until no valid takeover remains. If several takeovers are possible, they may be performed in any order.

For each query, determine how many corporations remain. Every query starts from the original arrangement. Takeovers in one query do not affect any other query.

Input

The first line contains two integers N and Q ($1 \le N, Q \le 2 \cdot 10^5$), the number of corporations and the number of queries.

The second line contains N integers $P_1, P_2, \ldots, P_N$ ($1 \le P_i \le 10^9$), the corporations' bank balances.

Each of the next Q lines contains two integers L and R ($1 \le L \le R \le N$), describing a query.

Output

For each query, print the number of remaining corporations on a separate line, in the order the queries appear in the input.

Sample Input 1	Sample Output 1
5 4 5 2 3 1 4 1 5 2 5 3 5 1 3	2 1 1 2

Problem F
Six-Seven-Seven-Six

Time limit: 4 seconds

The Council has convened, and your fate is yet unknown.
Solve the riddle laid before you, or never make it home.

N nodes, M edges in sight,
Q rounds, each a pair of nodes, a v and a u,
The distance is the fewest edges between the two.
Is it six-seven-seven-six or less?
The Council wants to know,
Are they close, or is the answer no?

Input

The first line contains three integers N, M and Q.

$$2 \leq N \leq 10^5$$
$$1 \leq M \leq \min(2 \times 10^5, N(N-1)/2)$$
$$1 \leq Q \leq 3 \times 10^5$$

The next $M + Q$ lines each contain two distinct integers u and v, where $1 \leq u, v \leq N$.

The first M lines describe distinct undirected edges. The remaining Q lines describe the queries.

Output

Print a binary string S of exactly Q characters, without spaces. Its i-th character is 1 if the distance between the vertices in query i is at most 6776, and 0 otherwise. Disconnected vertices give 0.

Sample Input 1	Sample Output 1
7 6 3 6 7 2 1 4 7 7 5 3 1 2 3 1 4 6 7 7 6	011

Problem G
Fab Fury's Therapy Session

Time limit: 2 seconds

Fab Fury is a psychologist who helps her patients manage recurring thoughts and mental pressure through a series of therapy sessions.

She represents a patient's initial mental pressure by a positive integer $\boldsymbol{M}$.

During each session, Fab Fury chooses a prime number $\boldsymbol{p}$ that divides the patient's current mental pressure $\boldsymbol{x}$. The session costs

$$\boldsymbol{x} \times \mathbf{d}(\boldsymbol{p}),$$

where $\mathbf{d}(\boldsymbol{p})$ is the number of decimal digits in $\boldsymbol{p}$.

After the session, the patient's mental pressure is divided by $\boldsymbol{p}$, reducing it from $\boldsymbol{x}$ to $\boldsymbol{x/p}$. The cost is calculated using the value of $\boldsymbol{x}$ **before this division**. A prime may be chosen again in a later session if it still divides the current mental pressure.

Fab Fury continues conducting sessions until the patient's mental pressure becomes $\mathbf{1}$. Given the initial mental pressure $\boldsymbol{M}$, determine the minimum possible total cost of all sessions.

For example, when $\boldsymbol{M} = \mathbf{330}$, choosing the primes in the order $\mathbf{5, 3, 11, 2}$ gives a total cost of

$$\mathbf{330 + 66 + 44 + 2 = 442}.$$

However, choosing them in the order $\mathbf{5, 3, 2, 11}$ gives

$$\mathbf{330 + 66 + 22 + 22 = 440},$$

which is the minimum possible cost.

Input

The first line contains an integer $\boldsymbol{T}$ ($\mathbf{1} \leq \boldsymbol{T} \leq \mathbf{100000}$), the number of test cases.

Each of the next $\boldsymbol{T}$ lines contains one integer $\boldsymbol{M}$ ($\mathbf{1} \leq \boldsymbol{M} \leq \mathbf{10^{12}}$), the patient's initial mental pressure.

The sum of $\lfloor\sqrt{\boldsymbol{M}}\rfloor$ over all test cases is at most $\mathbf{2 \times 10^7}$.

Output

For each test case, print one line containing the minimum total cost of all sessions.

Sample Input 1	Sample Output 1
4 1 330 12 999983	0 440 18 5999898

Problem H
Chotu ke Kanche

Time limit: 1 second

Chotu and his two friends have finished a game of kanche in the gali. Their three jars contain $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ kanche, respectively. Before going home, they want to share them equally.

Chotu offers to handle everyone's kanche himself. His friends don't trust him quite that much. Trust is important, but so is knowing where Chotu's hands are.

In one move, choose two different jars and transfer any positive integer number of kanche from one to the other. The source jar must contain enough kanche. Each transfer counts as **one move**, regardless of how many kanche are moved.

No kanche may be added or discarded. Jars may be empty.

Find the minimum number of moves needed to make all three jars contain the same number of kanche, or determine that it is impossible.

Input

The first line contains an integer $\boldsymbol{T}$ ($\mathbf{1} \leq \boldsymbol{T} \leq \mathbf{100000}$), the number of test cases.

Each of the next $\boldsymbol{T}$ lines contains three integers $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ ($\mathbf{0} \leq \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \leq \mathbf{10^8}$), the initial numbers of kanche in the jars.

Output

For each test case, print the minimum number of moves needed, or $-\mathbf{1}$ if the jars cannot be made equal.

Sample Input 1	Sample Output 1
6 10 10 10 8 10 12 2 2 17 1 2 17 0 0 0 0 0 3	0 1 2 -1 0 2

Problem I
One Road Down

Time limit: 2 seconds

A city has N junctions connected by M two-way roads. Each road has a fixed travel time in minutes, which is the same in both directions.

The ambulance station is at junction S, and the main hospital is at junction T. Ambulances always take the fastest available route from S to T.

The roads are old, so the city closes one road each night for repairs. Before deciding the repair schedule, the planners want to know how each closure would affect the ambulance service.

For each road independently, find the fastest travel time from S to T when **only that road is closed**. All other roads remain open.

Input

The first line contains four integers N, M, S and T ($2 \le N \le 100000$, $1 \le M \le 200000$, $1 \le S, T \le N$, $S \ne T$), the number of junctions, the number of roads, the ambulance station's junction and the hospital's junction.

Junctions are numbered from 1 to N.

Each of the next M lines contains three integers U, V and W ($1 \le U, V \le N$, $U \ne V$, $1 \le W \le 10^9$). They describe a two-way road between junctions U and V with a travel time of W minutes.

Roads are numbered from 1 to M in input order. There may be several roads between the same pair of junctions. These are separate roads, and closing one does not close the others.

When all roads are open, every junction is reachable from every other junction.

Output

Print M lines.

On line i, print the fastest travel time in minutes from S to T when road i is closed. Print -1 if the hospital cannot be reached during that closure.

Sample Input 1	Sample Output 1
7 9 1 7 1 2 2 2 3 2 3 6 2 1 4 3 4 3 2 2 5 4 5 6 3 4 5 6 6 7 1	8 8 10 7 7 7 7 7 -1

Problem J
Stenographer

Time limit: 2 seconds

Musaddiq has loved typing for as long as he can remember. As a child he raced his cousins on the family computer, and at university he became the fastest typist in his class. Now he has set his heart on becoming a court stenographer, so every evening he practises on an old stenotype machine he found at a market in Peshawar.

The machine has one key for every lowercase Latin letter. Pressing the key of letter c takes w_c units of effort. The machine is old, and some of its keys are jammed: for a jammed key $w_c = -1$, and that key cannot be pressed at all.

Like every stenotype, the machine also understands *chords*. There are m chords, and the j-th chord is described by three letters z_j, x_j and y_j. Musaddiq may choose to use this chord by first producing the letter x_j and then producing the letter y_j. The chord replaces these two results with a single letter z_j, with no additional effort. Each of x_j and y_j may itself be produced by pressing its key or by using a chord, so chords can be nested to any finite depth. Each chord may be used any number of times. The effort spent on a letter is the total effort of all key presses used to produce it.

The letters of s are produced independently, in order. Chords are applied only when Musaddiq chooses to use them, so completed letters of s are not automatically merged.

Tonight Musaddiq wants to type the string s. Help him find the minimum total effort needed to type it, or tell him that it cannot be typed on this machine.

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 26 integers $w_a, w_b, \ldots, w_z$ ($w_c = -1$ or $1 \le w_c \le 10^6$), the effort of pressing each key, or -1 if the key is jammed.

The second line contains a single integer m ($0 \le m \le 2 \cdot 10^5$), the number of chords.

Each of the next m lines contains three lowercase Latin letters z_j, x_j and y_j, separated by single spaces, describing a chord. The letters are not necessarily distinct, and the same chord may appear more than once.

The last line of the test case contains the string s ($1 \le |s| \le 2 \cdot 10^5$), consisting of lowercase Latin letters.

It is guaranteed that the sum of m over all test cases does not exceed $2 \cdot 10^5$, and the sum of $|s|$ over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer: the minimum total effort needed to type s, or -1 if some letter of s cannot be produced at all.

Sample Input 1

```
3
1 10 100 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
2
b a a
c b b
cab
1 1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0
add
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 7 3 4
3
x y z
x z z
x y y
xxy
```

Sample Output 1

```
7
-1
15
```

Explanation of Sample 1

In the first test case, the chord "b a a" produces "b" from two presses of "a". Using two separately produced copies of "b", the chord "c b b" produces "c". Producing "cab" this way takes $\mathbf{4} + \mathbf{1} + \mathbf{2} = \mathbf{7}$ units of effort.