Problem A. Haitang and Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given a set S, dXqwq and Haitang take turns performing the following operations, with dXqwq going first:

- Find a pair (x, y) such that $x, y \in S$ and $\gcd(x, y) \notin S$.
- Insert $\gcd(x, y)$ into S.

The player who cannot make a move loses the game. You need to output the winner when both players play optimally.

Input

Each test contains multiple test cases. The first line contains an integer T $(1 \le T \le 20)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$ — the size of set S.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(1 \le a_1 < a_2 < \cdots < a_n \le 10^5)$ — the elements of set S.

Output

For each test case, output "dXqwq" if dXqwq will win the game, and "Haitang" if Haitang will win the game.

Example

standard input	standard output
5 1 24 2 44 77 3 6 10 15 4 11 45 1419 19810 8 2 6 9 10 12 17 18 20	Haitang dXqwq Haitang dXqwq dXqwq

Problem B. Haitang and XOR Mex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

Haitang defines the mex of a non-negative integer sequence is the smallest non-negative integer that does not belong to the array. For example, $\mathrm{mex}([0,1,3]) = 2$.

Haitang defines the xormex of a non-negative integer sequence as the maximum value of the mex of the sequence after each element is XORed with the same non-negative integer. for example, $\mathrm{xormex}([8,9,11]) = \mathrm{mex}([8 \oplus 9, 9 \oplus 9, 11 \oplus 9]) = \mathrm{mex}([1,0,2]) = 3$.

Given a **permutation** a of length 2^n and m queries, each query consisting of two integers l_i and r_i, you need to calculate the sum of the xormex of the subarrays $[a_x, a_{x+1}, \cdots, a_y]$ for all $l_i \le x \le y \le r_i$.

Input

The first line contains two integers n and m ($1 \le n \le 18$, $1 \le m \le 10^6$) — the size of the permutation and the number of queries.

The second line contains 2^n integers a_i ($0 \le a_i < 2^n$) — the permutation a.

The next m lines each contain two integers l_i and r_i ($1 \le l_i \le r_i \le 2^n$) — the segment $[l_i, r_i]$ of this query.

Output

Output m lines, each containing an integer — the answer to i-th query.

Examples

standard input	standard output
2 4 3 2 0 1 1 3 2 3 1 2 1 4	9 3 4 19
3 5 0 4 6 7 5 2 1 3 1 8 3 5 2 6 3 7 1 4	93 9 29 22 15

Problem C. Haitang and Graph

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

Haitang calls a graph **perfect** if, for any pair of vertices (x, y) and for all $z \in [0, 2^w)$, there exists a path from x to y (not necessarily a simple path) such that the XOR sum of the weights of all edges on the path is equal to z. Specifically, if an edge is traversed multiple times, its weight is **counted multiple times** in the XOR sum.

A graph is simple if it does not contain multiple edges or self-loops.

Two graphs are considered different if there exists a pair of vertices (u, v) such that one graph has an edge (u, v) while the other does not, or if the weights of edge (u, v) are different in the two graphs.

Given four integers n, m, m_0, w, find the number of **perfect** simple undirected graphs with n vertices, m edges, edge weights in the range $[0, 2^w)$, with the given m_0 edges.

Print the answer modulo 998244353.

Input

The first line contains four integers n, m, m_0, w ($1 \le n \le 32$, $n-1 \le m \le \frac{n(n-1)}{2}$, $0 \le m_0 \le m$, $0 \le w \le 60$).

Each of the next m_0 lines contains three integers u_i, v_i, c_i ($1 \le u_i, v_i \le n$, $0 \le c_i < 2^w$), representing an edge.

It is guaranteed that the given edges form a simple graph.

Output

The only line contains an integer — the answer modulo 998244353.

Examples

standard input	standard output
3 3 1 1 1 2 1	2
4 6 0 3	86016
11 45 14 19 1 2 123 2 3 456 3 4 789 4 5 890 6 7 233 7 8 2333 8 9 23333 9 10 233333 6 8 333333 7 9 433333 8 10 0 11 1 4307 11 2 5307 11 3 6418	783514377

Problem D. Haitang and Uma Musume

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In this problem, you need to simulate the simplified calculation of training attributes in the game **Uma Musume Pretty Derby**.

Note that the rules in this problem are different from the original game.

Specifically, you will receive the information of an **Uma Musume** and 6 **Support Cards**, as well as several **trainings**, and you need to output the values of each attribute of the **Uma Musume**.

All **Uma Musume** have 6 abilities, which are **speed** $speed$, **stamina** sta, **power** $power$, **guts** $guts$, **wisdom** wis, and **skill points** $skill$.

An **Uma Musume** has 10 attributes, which are the initial values $speed_0, sta_0, power_0, guts_0, wis_0$ of the first 5 abilities and their training bonuses $speed_\times, sta_\times, power_\times, guts_\times, wis_\times$.

A **Support Card** has 13 attributes, including the friendship bonus $friend$, the motivation effect enhancement $drive_+$, the training effect enhancement $train$, the initial value bonus of the first 5 attributes $speed'_0, sta'_0, power'_0, guts'_0, wis'_0$, and the training bonus of the first 5 attributes $speed_+, sta_+, power_+, guts_+, wis_+$.

In a single game, you will choose an **Uma Musume** and 6 **Support Cards**. Then, you will let your **Uma Musume** do several **training**.

Each training consists of 6 attributes, including the training camp status $summer$, the overweight status $weight$, the current motivation $drive$, the type of training $type$, the level of training lv, and the set of support cards present (including whether each support card is in a friendship training state) S. lv will not be given directly, but the calculation method is as follows:

- During the training camp, lv is fixed at 5.
- Outside the training camp, lv starts at 1.
- After completing 4 times of the same type of training **outside** the camp, the lv of the **specific type** of training will increase by 1, but not exceeding 5.

In the following text, $\sum_{all}$ represents all carried support cards, $\sum_{present}$ represents all support cards present in the current training, and $\prod_{friends}$ represents all support cards present in the current training and in a friendship training state.

The initial number of skill points for the **Uma Musume** is 120, and the initial values of the remaining abilities are $\min(1200, X_0 + \sum_{all} X'_0)$, where X represents one among the 6 ability types.

The increase in an **Uma Musume**'s attribute after a training can be calculated according to the following formula, and we will provide a detailed explanation for each item:

$$\Delta X = \lfloor (base_{lv,type,X} + \textstyle\sum_{present} X_+) \cdot (\prod_{friends}(1 + 0.01 \cdot friend)) \cdot (1 + 0.01 \cdot \sum_{present} train) \\ \cdot (1 + coef_{drive} \cdot (1 + 0.01 \cdot \sum_{present} drive_+)) \cdot (1 + 0.01 \cdot X_\times) \cdot (1 + \sum_{present} 0.05) \rfloor$$

- Base training value $(base_{lv,type,X} + \sum_{present} X_+)$

The $base_{lv,type,X}$ array will be provided in the appendix. Specifically, this represents the base value for the increase of each ability, for each type of training, at each level.

$\sum_{present} X_+$ represents the sum of the attribute bonuses of all support cards present, and $skill_+$ is always equal to zero.

- Friendship bonus ($\prod_{friends}(1 + 0.01 \cdot friend)$)

For each **Support Card** with a **friendship training**, we add one to its friendship bonus and then multiply them all together.

- Training effect enhancement ($1 + 0.01 \cdot \sum_{present} train$)

We sum up the training bonuses of all **Support Cards** present and add one.

- Motivation bonus ($1 + coef_{drive} \cdot (1 + 0.01 \cdot \sum_{present} drive_{+})$)

The motivation ranges from $[0, 4]$, and $coef_i$ are respectively $-0.2, -0.1, 0, 0.1, 0.2$.

For each **Support Card** present, we sum up their motivation bonuses and add one, then multiply by the motivation coefficient.

- **Uma Musume** growth rate ($1 + 0.01 \cdot X_{\times}$)

This parameter is only related to the **Uma Musume**'s attributes, and $skill_{\times}$ is always equal to zero.

- Number of support cards present ($1 + \sum_{present} 0.05$)

Each present **Support Card** will provide a 0.05 bonus.

- Other rules

In particular, if $weight = 1$, the $\Delta speed$ will be forced to become 0, i.e. the speed cannot be increased.

After training, if the first 5 abilities exceed 1200, we will take the minimum of 1200, note that this rule does not apply to skill points.

You need to print the 6 abilities of the **Uma Musume** after each training.

Input

The first line contains 10 integers $speed_0, sta_0, power_0, guts_0, wis_0, speed_{\times}, sta_{\times}, power_{\times}, guts_{\times}, wis_{\times}$ ($50 \le speed_0, sta_0, power_0, guts_0, wis_0 \le 200$, $0 \le speed_{\times}, sta_{\times}, power_{\times}, guts_{\times}, wis_{\times} \le 30$), representing the **Uma Musume**.

Each of the next 6 lines contains 13 integers
$friend, drive_{+}, train, speed'_0, sta'_0, power'_0, guts'_0, wis'_0, speed_{+}, sta_{+}, power_{+}, guts_{+}, wis_{+}$
($0 \le friend, drive_{+}, train, speed'_0, sta'_0, power'_0, guts'_0, wis'_0, speed_{+}, sta_{+}, power_{+}, guts_{+}, wis_{+} \le 30$), representing a **Support Card**.

The next line contains an integer n ($1 \le n \le 72$) — the number of **trainings**.

Each of the next n lines begins with 5 integers $summer, weight, drive, type, |S|$ ($0 \le summer, weight \le 1$, $0 \le drive, type \le 4$, $0 \le |S| \le 5$), representing a **training**. Then $|S|$ pairs of integers x_i, y_i ($1 \le x \le 6$, $0 \le y \le 1$) follow, representing the x_i-th **Support Card** is present in this training, and $y = 1$ means it has the **friendship training**.

It is guaranteed that in each training, x_i are pairwise different.

Output

For each test case, print one line contains 6 integers — the abilities of the **Uma Musume**.

Example

standard input	standard output
71 117 70 102 90 0 10 0 20 0 25 30 15 0 0 0 0 0 0 1 0 0 0 30 30 15 0 0 0 0 30 0 0 0 0 1 25 0 15 0 0 0 0 0 0 0 0 0 2 30 30 10 0 0 20 0 0 0 0 1 0 0 30 30 0 0 0 0 20 0 1 0 1 0 0 30 0 10 0 0 0 0 0 0 0 0 1 0 9 0 1 0 0 1 5 0 0 0 1 0 4 1 1 2 0 3 0 4 0 0 1 2 0 3 1 0 3 0 6 0 0 0 3 0 5 1 1 2 1 3 1 4 1 5 1 0 1 4 0 3 2 1 4 0 6 1 1 0 3 1 0 1 1 4 2 1 3 1 1 0 3 3 3 4 1 5 0 6 1 1 1 4 4 1 4 0	71 117 94 122 120 121 89 119 105 122 125 124 89 120 113 123 128 127 178 128 169 123 152 143 178 128 189 127 155 149 178 143 189 132 155 151 178 156 210 132 158 154 194 156 228 174 158 159 194 156 229 174 176 164

Note

Here is the table of $base$:

Table $base_1$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	10	0	5	0	0	2
1	0	9	0	4	0	2
2	0	5	8	0	0	2
3	4	0	4	8	0	2
4	2	0	0	0	9	4

Table $base_2$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	11	0	5	0	0	2
1	0	10	0	4	0	2
2	0	5	9	0	0	2
3	4	0	4	9	0	2
4	2	0	0	0	10	4

Table $base_3$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	12	0	5	0	0	2
1	0	11	0	4	0	2
2	0	6	10	0	0	2
3	4	0	4	10	0	2
4	3	0	0	0	11	4

Table $base_4$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	13	0	5	0	0	2
1	0	12	0	4	0	2
2	0	6	11	0	0	2
3	4	0	4	11	0	2
4	3	0	0	0	12	4

Table $base_5$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	14	0	5	0	0	2
1	0	13	0	4	0	2
2	0	7	12	0	0	2
3	5	0	5	12	0	2
4	4	0	0	0	13	4

The author set this problem after he had uninstalled the game.

Problem E. Haitang and Math

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Haitang defines the $S(m)$ of a positive integer m as the sum of digits in m.

For example, $S(154) = 1 + 5 + 4 = 10$, $S(147) = 1 + 4 + 7 = 12$.

Given a positive integer n, count the number of positive integers $m \leq n$ such that $n \bmod m = S(m)$.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \leq T \leq 100$) — the number of test cases. The description of the test cases follows.

The first and only line of each test case contains an integer n ($1 \leq n \leq 10^{12}$).

Output

For each test case, print one line containing an integer — the answer.

Example

standard input	standard output
8 11 45 14 191 9810 20061130 20080520 998244353	1 3 0 3 17 10 3 15

Problem F. Haitang and Diameters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Haitang defines the **distance** between a pair of points on a tree as the sum of the weights of all the edges on the simple path.

Haitang defines the number of diameters of a tree as the number among $\binom{n}{2}$ ways to choose a pair of points with the maximum **distance**.

Given a tree with n vertices, you can assign a weight of 0 or 1 to each edge.

You need to find the sum of the number of diameters of all the trees generated by all 2^{n-1} assignment methods, modulo 998244353.

Input

The first line contains a single integer n ($2 \le n \le 2000$) — the number of vertices in the tree.

Each of the next $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), representing an edge between u_i and v_i.

It is guaranteed that the given edges form a tree.

Output

The only line contains an integer — the answer modulo 998244353.

Examples

standard input	standard output
3 1 2 2 3	8
5 1 2 1 3 2 4 2 5	50

Problem G. Haitang and Rock Paper Scissors

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Haitang is playing a Rock Paper Scissors match with n rivals in a row.

Before the match starts, Haitang can predict the shapes they form as a sequence a, where $a_i = 0, 1, 2$ indicates rock, paper, and scissors. However, she can't form the same shape in any two contiguous games.

In a match containing n games, the total score is calculated as follows:

- When Haitang wins the game, she gains 1 point.
- When the game is tied, nothing happens.
- When Haitang loses the game, she loses everything and the whole match ends immediately.

You are given a sequence b of length n, where $b_i \in \{-1, 0, 1, 2\}$. There are $3^{\text{number of -1}}$ ways to replace each -1 by $0, 1, 2$ to generate the sequence a.

Haitang will play all matches with every possible a. You need to print the sum of the maximum points Haitang could gain in each match.

Print the answer modulo 998244353.

Input

The first line contains an integer n ($1 \le n \le 2000$) — the length of sequence b.

The second line contains n integers b_i ($b_i \in \{-1, 0, 1, 2\}$).

Output

The only line contains an integer — the answer modulo 998244353.

Examples

standard input	standard output
4 -1 0 1 2	11
7 0 -1 1 2 -1 1 0	49

Problem H. Haitang and Water

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

VERTeX (rintaro soma deconstructed remix)
— Rintaro Soma

There are n bottles lined up, numbered $1, 2, \cdots, n$, with the volume of the i-th bottle being a_i. Initially, the first bottle is full of water, and the rest are empty.

Haitang performs the following operations on each bottle in order from left to right:

- Select a bottle uniformly at random from the remaining $n-1$ bottles.
- Pour the water from the current bottle into the selected bottle until the current bottle is empty or the selected bottle is full.

Given the sequence a, dXqwq wants to know the expected amount of water in each bottle after all operations.

Since dXqwq dislikes floating-point numbers, you only need to print the result modulo 998244353.

Input

The first line contains an integer n ($2 \le n \le 10^5$) — the number of bottles.

The second line contains n integers a_i ($1 \le a_i \le n$) — the volume of each bottle.

Output

Print n lines, where the i-th line contains the value of the expected volume of water in the i-th bottle after all operations, modulo 998244353.

Examples

standard input	standard output
2 1 1	1 0
3 3 1 2	623902723 623902721 748683265
9 9 9 8 2 4 4 3 5 3	304464287 164086171 361005467 588475930 898938779 983453531 155241138 69810681 467501437

Problem I. Haitang and Ranking

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

In this problem, a ranking does not have any ties (i.e. a ranking should be a permutation of $1 \sim n$).

n contestants participated in two competitions. In each competition, the ranking of the i-th contestant is a_i and b_i respectively.

dXqwq received a total ranking from Haitang, where the ranking of the i-th contestant is c_i.

Obviously, if a contestant's ranking is higher than another contestant's in both competitions, his total ranking should also be higher. dXqwq wants you to check if c satisfies this condition.

Since dXqwq has found a contradiction in the ranking, Haitang makes m operations. In each operation, two integers x, y will be given, and Haitang will swap c_x, c_y. You need to determine whether c satisfies this condition after each operation.

Notice that operations are **not** independent.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 10^5$) — the number of the permutation and the number of queries.

The second line contains n integers a_i ($1 \leq a_i \leq n$) — the permutation a.

The third line contains n integers b_i ($1 \leq b_i \leq n$) — the permutation b.

The fourth line contains n integers c_i ($1 \leq c_i \leq n$) — the permutation c.

The next m lines each contain two integers x and y ($1 \leq x, y \leq n$), representing an operation.

Output

After each operation, output "Yes" if the ranking satisfies the condition, and "No" otherwise.

Example

standard input	standard output
4 4 1 2 3 4 1 3 2 4 2 1 3 4 1 3 1 2 2 3 3 4	No Yes Yes No

Problem J. Haitang and Triangle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given two integers n, m, construct a permutation of length n that satisfies the following conditions.

- There are exactly m subintervals of length 3 such that the numbers in these subintervals form a (non-degenerate) triangle.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \le T \le 10^5$) — the number of test cases. The description of the test cases follows.

The first and only line of each test case contains two integers n and m ($3 \le n \le 3 \times 10^5$, $0 \le m \le n - 2$) — the length of permutation and the target subintervals.

It is guaranteed that the sum of n over all test cases does not exceed 3×10^5.

Output

For each test case, print one line.

If such a pair of permutations exists, print n integers p_i, representing the permutation you have constructed. Otherwise, print "-1".

Example

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

Problem K. Haitang and Ava

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

One day, Haitang found that Ava would not hold live streaming anymore.

Ava would say an opening statement at the beginning of the live streaming.

The conditions for a valid opening statement are as follows:

- An empty string is a valid opening statement.
- If S is a valid opening statement, then $S + \texttt{ava}$ and $\texttt{ava} + S$ are also valid opening statements.
- If S is a valid opening statement, then $S + \texttt{avava}$ and $\texttt{avava} + S$ are also valid opening statements.
- Any string that cannot be constructed using the above methods is not a valid opening statement.

Given a string S, you need to determine if it is a valid opening statement.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \le T \le 1.7 \times 10^5$) — the number of test cases. The description of the test cases follows.

The first and only line of each test case contains a string S ($3 \le |S| \le 5 \times 10^5$), consisting of lowercase letters of the English alphabet.

It is guaranteed that the sum of $|S|$ over all test cases does not exceed 5×10^5.

Output

For each test case, output "Yes" if S is a valid opening statement, and "No" otherwise.

Example

standard input	standard output
5 ava avavaava avavava avaava haitang	Yes Yes No Yes No