Problem A. Swapping Inversions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given a permutation x of the integers from 1 to n.

You want to sort this permutation by a sequence of operations. In one operation, you select two adjacent elements x_i and x_{i+1} such that $x_i > x_{i+1}$ and swap them. When there are multiple choices of such i, you choose one of them with equal probability. When there is no such i, the process ends.

The cost of swapping x_i and x_{i+1} is $|x_i - x_{i+1}|$. Calculate the expected total cost of sorting the permutation modulo $10^9 + 7$.

Input

The first line of input contains an integer n ($1 \le n \le 10^6$).

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le n$). It is guaranteed that x is a permutation of the integers from 1 to n.

Output

Print a single line containing an integer: the expected total cost modulo $10^9 + 7$.

Formally, it can be shown that the expected total cost can be represented as a fraction p/q for some coprime non-negative integers p and q. For example, if the expected total cost is an integer, then we just have $q = 1$. You have to print the value $p \cdot q^{-1} \bmod (10^9 + 7)$.

Examples

standard input	standard output
5 1 2 3 4 5	0
5 1 2 5 3 4	3

Problem B. Hamiltonian Path

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a directed graph of n vertices numbered from 0 to $n-1$. You are also given two integers p and q such that $1 \le p, q \le n$.

The edges of the graph are constructed as follows: for every vertex i,

- if $i + p < n$, then there is an edge from i to $i + p$;
- if $i - q \ge 0$, then there is an edge from i to $i - q$.

Obviously, the graph has exactly $(n-p) + (n-q)$ edges.

Find any Hamiltonian path in this graph, or determine that it does not exist.

Recall that a Hamiltonian path is a path that visits every vertex exactly once.

Input

The first line of input contains an integer T ($1 \le T \le 10^4$), the number of test cases.

Each test case consists of a single line containing three integers: n, p, and q ($1 \le p, q \le n \le 10^6$).

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

Output

For each test case, print a single line containing n integers that represent the order of vertices in a Hamiltonian path, or print -1 if it does not exist.

If there are multiple solutions, print any one of them.

Example

standard input	standard output
3 5 3 2 8 2 4 13 5 7	2 0 3 1 4 -1 0 5 10 3 8 1 6 11 4 9 2 7 12

Problem C. Minimal Cyclic Shift

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

Ani is a young and reckless student. One day, he got a really weird math homework.

In the homework, he was given n strings $s_1, s_2, \ldots, s_n$ with length $a_1, a_2, \ldots, a_n$, respectively.

Define $f(s)$ as the position where the lexicographically minimal cyclic shift of s starts. Since it may not be unique, $f(s)$ is defined as the minimal such position. For example, $f($"`qweqweqwe`"$) = 3$, because the lexicographically minimal cyclic shift of $s =$ "`qweqweqwe`" is "`eqweqweqw`", and the minimal possible position where it starts in s is position 3 where the first letter "`e`" is located.

The homework was to write down $f(s_1), f(s_2), \ldots, f(s_n)$, in this order. But Ani's recklessness and the approaching of the deadline caused him to write the answers in the order $f(s_n), f(s_1), \ldots, f(s_{n-1})$.

Ani had not realized this until he submitted his answers. Now he can only remember $a_1, a_2, \ldots, a_n$. Assuming the given strings contain only lowercase English letters and were generated uniformly at random by the teacher, you need to help him calculate the expected number of correct answers in his homework modulo 998 244 353.

Input

The first line of input contains an integer n ($1 \le n \le 10^5$), the number of strings given in Ani's homework.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) separated by spaces, indicating the lengths of the strings.

Output

Print a single line with an integer: the expected number of correct answers in Ani's homework modulo 998 244 353.

Formally, it can be shown that the expected number of correct answers can be represented as a fraction p/q for some coprime non-negative integers p and q. You have to print the value $p \cdot q^{-1} \bmod 998\,244\,353$.

Examples

standard input	standard output
5 3 1 5 2 4	727907401
1 100000	1

Problem D. Interval

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are given n intervals, the j-th of which is $I_j = [l_j, r_j]$.

Define the beauty of $[L, R]$ as the length covered by $\bigcup_{i=L}^{R} [l_i, r_i]$.

You are given m queries, the i-th of which is $[A_i, B_i]$, and you need to answer:

If we uniformly sample $[L_i, R_i]$ from all possible integer pairs such that $A_i \le L_i \le R_i \le B_i$, what is the expected value of the beauty of $[L_i, R_i]$?

Find the answers modulo 998 244 353.

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$).

Each of the following n lines contains two integers l_j and r_j ($0 \le l_j < r_j \le 10^8$).

Each of the following m lines contains two integers A_i and B_i ($1 \le A_i \le B_i \le n$).

Output

Output m lines, each of which contains the answer for a query modulo 998 244 353.

Formally, it can be shown that the expected beauty can be represented as a fraction p/q for some coprime non-negative integers p and q. You have to print the value $p \cdot q^{-1} \bmod 998\,244\,353$.

Example

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

Note

The size of input and output is large. Remember to use fast input and output methods to avoid getting a “Time Limit Exceeded”.

Problem E. PlayerUnknown's Battlegrounds

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

When Rikka was playing PUBG (PlayerUnknown's Battlegrounds), she could not win the game. So she decided to use some techniques this time. With the techniques, now Rikka can see clearly where all players are in the game.

In the game, the map can be denoted as a grid of size $n \times m$. Rikka uses software that can rate the enemy's combat value based on their performance. Now, there is exactly one enemy in each square, and all enemies' combat values form a permutation of integers from 1 to $n \cdot m$. Since Rikka is quite a green hand, she always cares about the weakest enemies (the less the enemy's combat value is, the weaker the enemy is).

We denote the square at row i, column j as (i, j). A subgrid can be defined by $((x_1, y_1), (x_2, y_2))$, where $1 \le x_1 \le x_2 \le n$ and $1 \le y_1 \le y_2 \le m$, and the subgrid itself is the squares (x, y) for which $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$. Two subgrids $((x_1, y_1), (x_2, y_2))$ and $((x'_1, y'_1), (x'_2, y'_2))$ are the same if and only if $((x_1, y_1), (x_2, y_2)) = ((x'_1, y'_1), (x'_2, y'_2))$.

For now, Rikka wants to know how many subgrids are there such that the weakest enemy's combat value in them is equal to x. Help her find the answers for all $x = 1, 2, \ldots, n \cdot m$.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 300$).

Then follow n lines. The i-th of them contains m integers, where the j-th integer is the combat value of the enemy in (i, j). It is guaranteed that the values form a permutation of integers from 1 to $n \cdot m$.

Output

Print $n \cdot m$ lines, with one integer on each line.

The integer on the i-th line is the answer for $x = i$.

Example

standard input	standard output
2 3 2 5 1 6 3 4	6 4 5 1 1 1

Problem F. Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Given a rectangular array a of size $n \times m$ and a prime number p, find two rectangular arrays, b of size $K \times n$ and c of size $K \times m$, such that:

1. $0 \le b_{i,j} < p$ $(\forall 1 \le i \le K, 1 \le j \le n)$;
2. $0 \le c_{i,j} < p$ $(\forall 1 \le i \le K, 1 \le j \le m)$;
3. $\sum_{j=1}^{n} b_{i,j} \ge 1$ $(\forall 1 \le i \le K)$;
4. $\sum_{j=1}^{m} c_{i,j} \ge 1$ $(\forall 1 \le j \le K)$;
5. $\sum_{l=1}^{K} b_{l,i} \cdot c_{l,j} \equiv a_{i,j} \pmod p$ $(\forall 1 \le i \le n, 1 \le j \le m)$.

Input

The first line of input contains four integers n, m, K, p ($1 \le n \cdot m, K \cdot n, K \cdot m \le 10^5$; $2 \le p \le 10^9 + 7$; p is prime).

The i-th of the following n lines contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ $(0 \le a_{i,j} < p)$.

Output

If there is no solution, output a line "`No solution!`".

Otherwise, output K lines, i-th of which contains $n + m$ integers $b_{i,1}, b_{i,2}, \ldots, b_{i,n}, c_{i,1}, c_{i,2}, \ldots, c_{i,m}$.

If there are several possible answers, print any one of them.

Examples

standard input	standard output
1 1 1 97 0	No solution!
3 3 1 97 1 2 3 2 4 6 3 6 9	1 2 3 1 2 3

Problem G. Reasonable Workplace Relationship

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

The subordinates must obey the orders of the superiors unconditionally. If there is any objection, they can say it later. "If you throw the pot up and down to the subordinates, they can only accept it with patience and full of grievances"... Do these words often seen in the workplace novels really reflect the current situation? Let us turn to a more specific simplified model.

Now, we have a workplace relationship model, which is a superior–subordinate relationship. Let us assume that there is only one big boss. Since they are called the big boss, they naturally have no superior. The big boss has some subordinates who are directly managed by them, and these subordinates have their own subordinates... Repeat several times like this, and we can clearly see that this model is a rooted tree. There are n people in the model, numbered from 1 to n. The superior of person i is the parent of node i, and the big boss is the root of the tree. We will give each person i an integer a_i to measure their ability.

A *group* consists of a person, which is called the *leader* of the group, along with all their direct and indirect subordinates. Clearly, a group corresponds to some subtree, and the group leader is the root of that subtree. We know that the management of subordinates by leaders should not be based only on the ability of leaders and subordinates, which is too narrow and not conducive to management. Because of their special status as leaders, leaders obviously still have the so-called prestige to help them manage. So we will give each person i an integer w_i to measure their prestige.

With prestige and certain ability, leadership can convince the public. Unfortunately, there are always some subordinates whose ability value (a_i) is greater than the sum of the leader's ability and prestige, and this is very bad. No matter how open-minded a leader is, there will always be some discomfort in their heart.

In order to simplify the problem, consider a group with leader i. Let the subtree rooted at i have s_i nodes. Let k_i be the number of such nodes j in this subtree that satisfy $a_j > a_i + w_i$. Then person i has the probability of k_i/s_i of becoming unhappy as a leader (leaders are simple, they are either happy or unhappy). The probabilities for different people are independent.

Now, in order to measure the reasonableness of the company's workplace relationship structure, let us look at some groups. Specifically, we have to answer m questions. For question i, consider the group led by person x_i. We have to count the expected number of people in this group that will be happy as leaders.

Input

The first line of input contains two integers n and m ($2 \le n \le 3 \cdot 10^5$, $1 \le m \le 3 \cdot 10^5$).

The next line contains n integers $p_1, p_2, \ldots, p_n$, where p_i denotes the superior of person i. If $p_i = 0$, the i-th person has no superior, and is the big boss. It is guaranteed that there is exactly one big boss, but it is **not** guaranteed that the boss is the person number 1. It is also guaranteed that the superior–subordinate relationships form a rooted tree.

The following line contains n integers $a_1, a_2, \ldots, a_n$ denoting the ability of each person ($1 \le a_i \le 10^9$).

The subsequent line contains n integers $w_1, w_2, \ldots, w_n$ denoting the prestige of each person ($1 \le w_i \le 10^9$).

The i-th of the following m lines contains one integer x_i, asking how many people in the group led by x_i are expected to be happy.

Output

Print m lines, with one integer on each: the answers to the queries.

Since the result of a query may not be an integer, you need to output the values modulo $10^9 + 7$.

Formally, it can be shown that the answer can be represented as a fraction p/q for some coprime non-negative integers p and q. You have to print the value $p \cdot q^{-1} \bmod (10^9 + 7)$.

Example

standard input	standard output
2 2 0 1 1 10 1 1 1 2	500000005 1

Note

Person 1 will be happy with probability 1/2. Person 2 will always be happy.

Problem H. Historic Breakthrough

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Rikka is taking the undergraduate cryptography course this year. In the class, it is mentioned that a new method of factorization was recently discovered, which is considered a historic breakthrough in cryptography. Rikka believes that there will be comparable algorithms for some specific class of numbers with simpler goals.

Now you are asked to design and implement such a "factorization" algorithm: given an integer m, print an integer n such that

$$m = \frac{n\varphi(n)}{2} = \sum_{\substack{1 \le i \le n,\\ \gcd(i,n)=1}} i.$$

It is guaranteed that such an n exists in all the test cases.

Input

The first line of input contains an integer T ($1 \le T \le 51$), the number of test cases.

The i-th of the following T lines contains a single integer m_i ($0 < m_i < 10^{36}$).

Output

For each test case, print an integer n_i for the corresponding m_i. It is guaranteed that the answer exists in every test case. If there are several possible answers, print any one of them.

Example

standard input	standard output
3 20 21 475750381222420656586462245096576000	10 7 1497700821900508526

Problem I. Nondeterministic Finite Automaton

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

A *nondeterministic finite automaton* (NFA) is defined as $G = (V, E_0, E_1, v_0, F)$, where (V, E_0) and (V, E_1) form two directed graphs, $v_0 \in V$ is the initial vertex, and $F \subseteq V$ is the set of accepting vertices.

We say that NFA G recognizes a 01-string $s = s_1 s_2 \ldots s_n$ if and only if there exists a sequence of vertices $u_0, u_1 \ldots u_n$ such that $u_0 = v_0$, the edges $\langle u_{i-1}, u_i \rangle \in E_{s_i}$ for all $i = 1, 2, \ldots, n$, and $u_n \in F$.

Define $L = L(G)$ as the minimal non-negative integer such that there exists a string s of length L which G can not recognize. If no such L exists for G, we define $L(G) = -1$.

You are given n, and you need to construct an NFA $G = (V, E_0, E_1, v_0, F)$ such that $|V| = n$ and $L(G)$ is large enough. The exact constraints on n and $L(G)$ are at the bottom.

Input

The first line of input contains an integer n.

Output

Output the NFA G you found.

Suppose the vertices in V are labeled by integers $0, 1, \ldots, n-1$.

Firstly, output E_0 in the following format: The first line contains an integer $e = |E_0|$ $(0 \le e \le 1000)$. Then e lines follow. The i-th of them contains two integers x_i and y_i $(0 \le x_i, y_i < n)$, indicating that there is an edge $\langle x_i, y_i \rangle \in E_0$. Note that $x_i = y_i$ is allowed.

Secondly, print E_1 in the same format as E_0.

Next, print a line with the integer k.

After that, print a line containing k integers $f_1, f_2, \ldots, f_k$, indicating that $F = \{f_1, f_2, \ldots, f_k\}$.

Example

Below is an example that **does not** appear in the tests. Here, $L(G) = 4$, because G cannot recognize the string "`1010`".

standard input	standard output
3	2 0 0 2 2 4 0 1 1 0 0 2 2 1 3 0 1 2

Note

This problem has two tests: $n = 6$ and $n = 20$.

When $n = 6$, your output's $L(G)$ should be strictly greater than 18.

When $n = 20$, your output's $L(G)$ should be strictly greater than 400.

Problem J. Texas Hold ’em

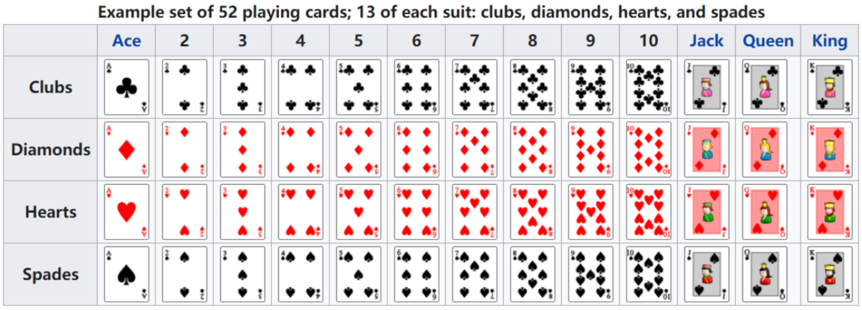
Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	256 mebibytes

Texas hold ’em (also known as Texas holdem, hold ’em, and holdem) is one of the most popular variants of the card game of poker. **Please read the following rules as they may be different from the regular rules.**

Now, two players, Alice and Bob, are playing Texas holdem. At the beginning of each game, each player keeps two cards, known as hole cards. In this problem, different from the regular Texas holdem, where the hole cards are put face-down, **both players know all the four hole cards**. Then, five community cards are dealt in three stages face-up. The stages consist of a series of three cards (“the flop”), later an additional single card (“the turn” or “fourth street”), and a final card (“the river” or “fifth street”). All players know the face-up cards that are already dealt.

All cards are drawn from a standard 52-card deck. A standard 52-card deck comprises 13 ranks in each of the four French suits: clubs (♣), diamonds (♢), hearts (♡), and spades (♠). Each suit includes an Ace (A), a King (K), Queen (Q) and Jack (J), each depicted alongside a symbol of its suit; and numerals or pip cards from the Deuce (Two) to the Ten, with each card depicting that many symbols (pips) of its suit. No card can be drawn more than once.

Example set of 52 playing cards; 13 of each suit: clubs, diamonds, hearts, and spades

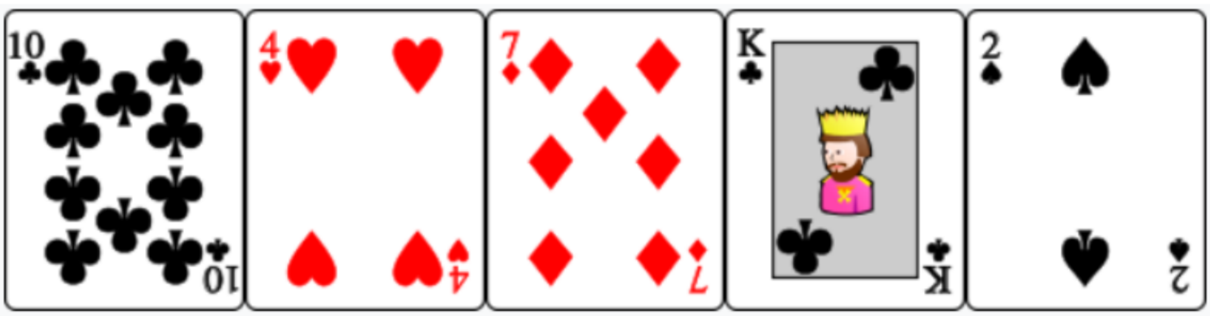


Individual cards are ranked as follows (high-to-low): Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2. Each player seeks the best five-card poker hand from any combination of the seven cards — the five community cards and the player’s two hole cards.

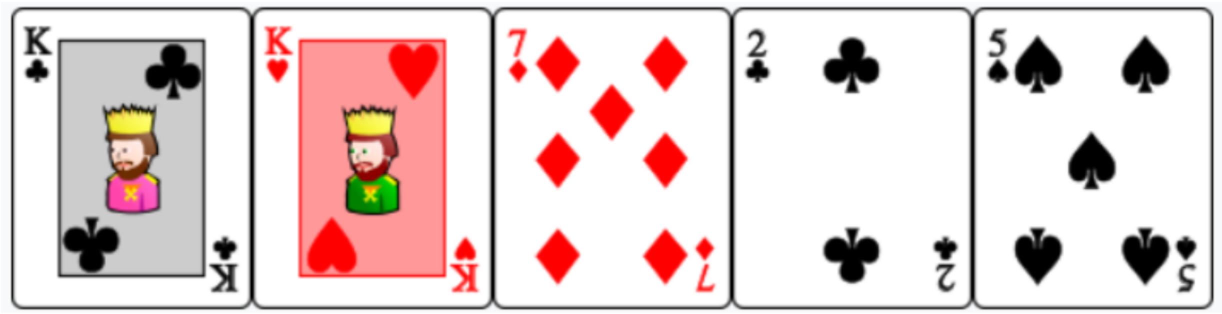
The following table shows the possible five-card poker hand types in increasing order of their values. Each type has a specific ordering of the five cards that are described below. The following part is describing how to compare two hands, which is the same as the regular rule.

- **Highcard**: Simple value of the card. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 > a_2 > a_3 > a_4 > a_5$.

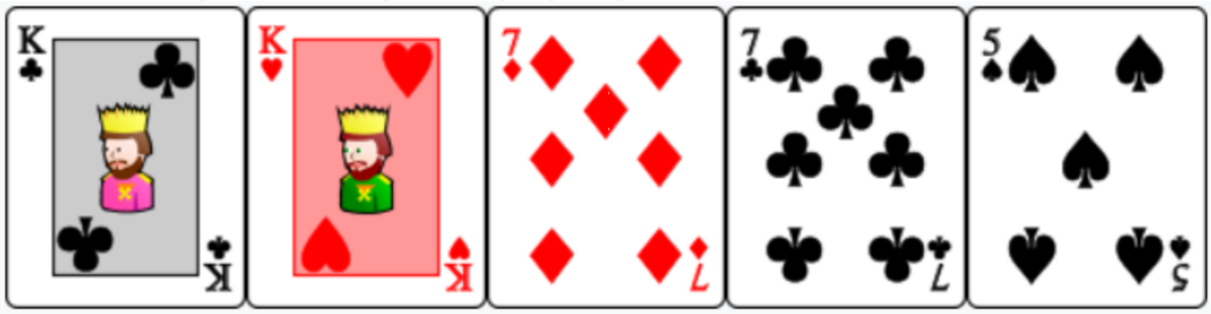
  Here and below, a_i represents the rank of i-th card. Note that, while the pictures here and below show examples of hand types, the cards on the pictures may not yet be ordered properly.



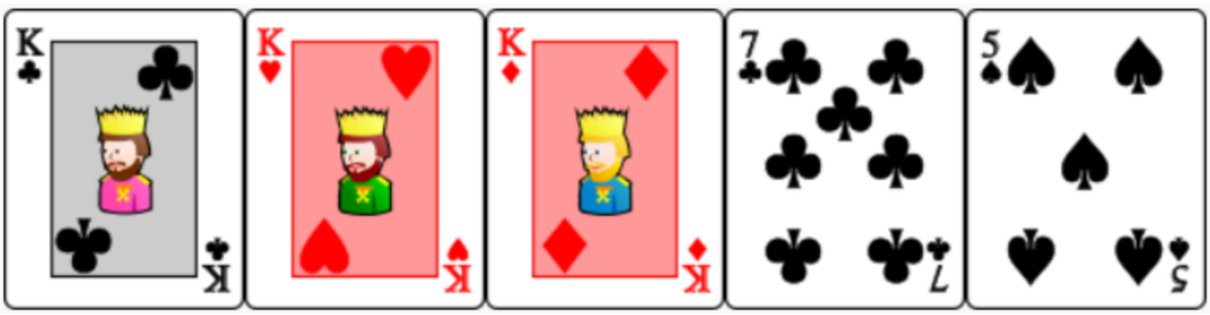
- **Pair**: Two cards with the same value. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 = a_2$, $a_3 > a_4 > a_5$, $a_1 \neq a_3$, $a_1 \neq a_4$, and $a_1 \neq a_5$.



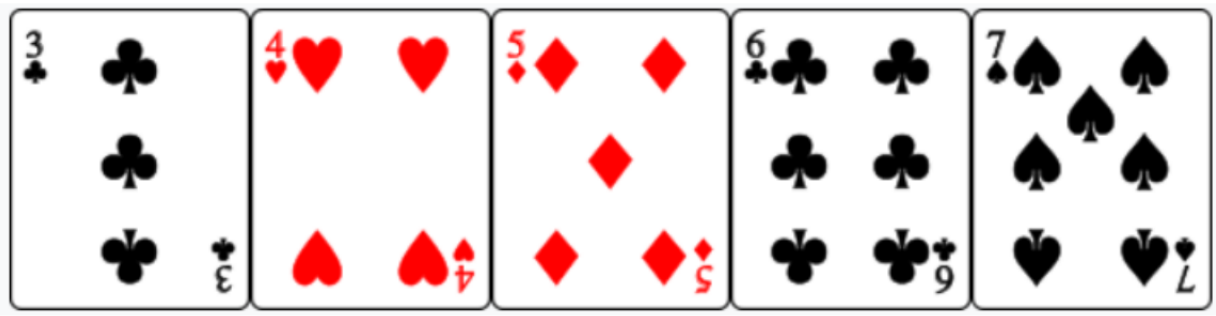
- **Two pairs**: Two times two cards with the same value. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 = a_2$, $a_3 = a_4$, $a_1 > a_3$, $a_1 \neq a_5$, and $a_3 \neq a_5$.



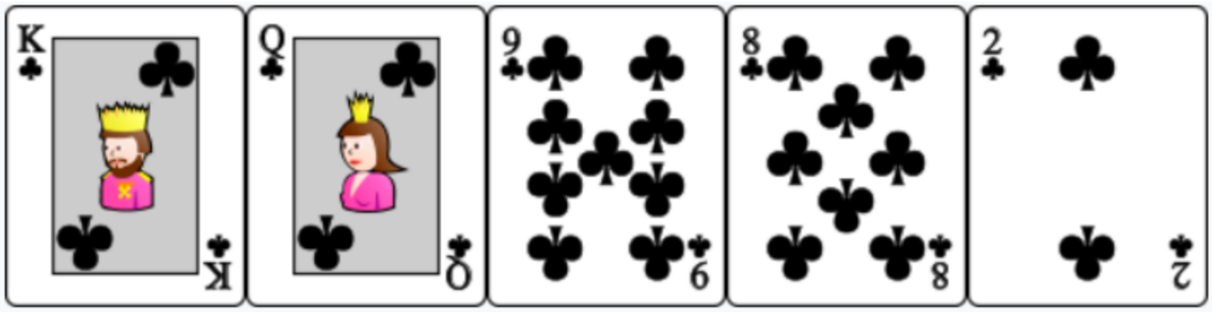
- **Three of a kind**: Three cards with the same value. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 = a_2 = a_3$, $a_4 > a_5$, $a_1 \neq a_4$, and $a_1 \neq a_5$.



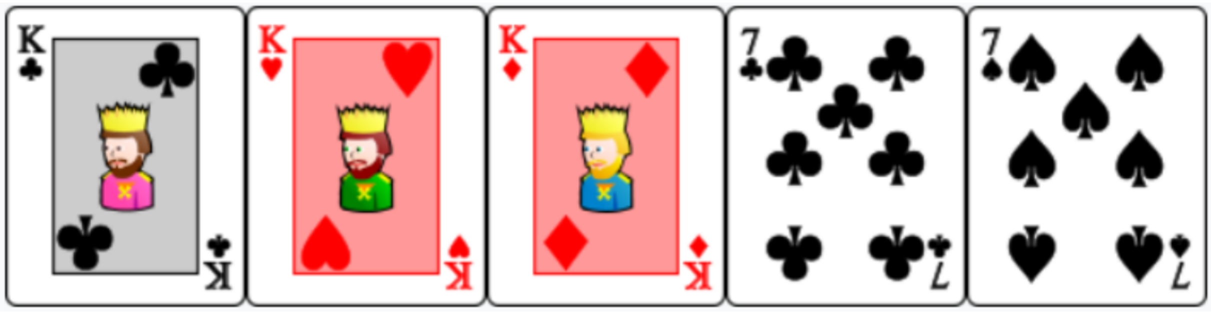
- **Straight**: Sequence of five cards in increasing value. The cards are ordered as $a_1a_2a_3a_4a_5$ such that a_i is exactly one rank above a_{i+1} for all $1 \leq i \leq 4$. As a special case, if a_5 is Ace, a_4 can be 2. In this case, Ace is considered one rank below 2.



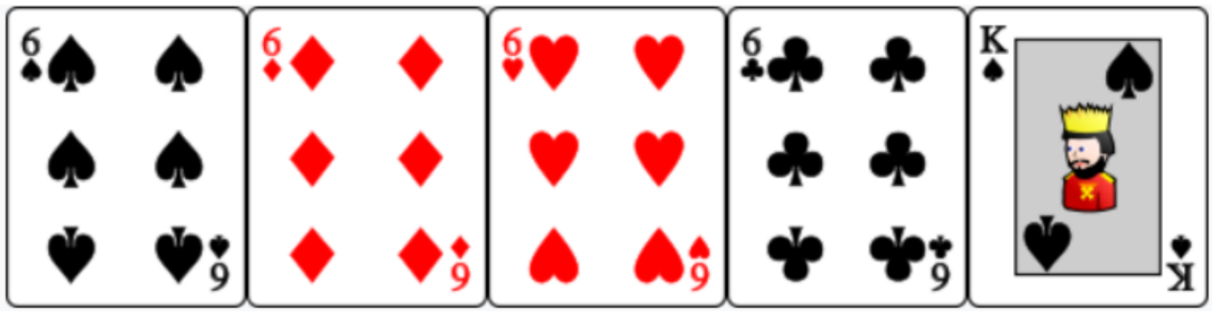
- **Flush**: Five cards of the same suit. The cards are ordered as $a_1a_2a_3a_4a_5$ such that all the five cards have the same suit and $a_1 > a_2 > a_3 > a_4 > a_5$.



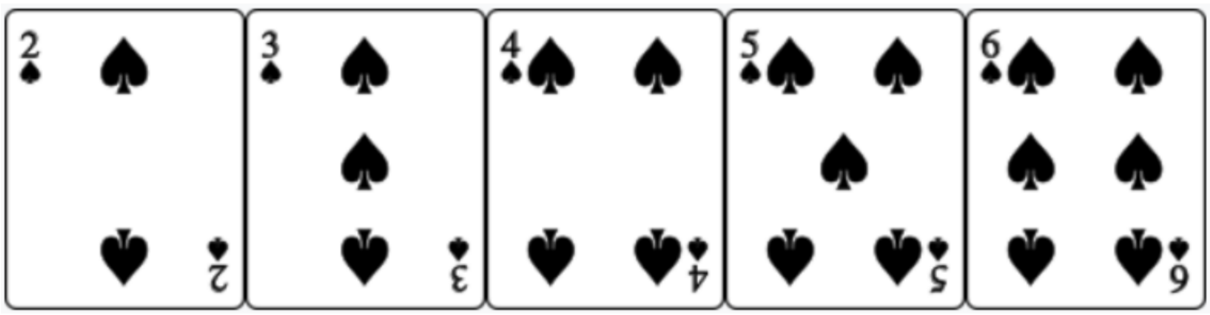
- **Full house**: Combination of three of a kind and a pair. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 = a_2 = a_3$ and $a_4 = a_5$.



- **Four of a kind**: Four cards of the same value. The cards are ordered as $a_1a_2a_3a_4a_5$ such that $a_1 = a_2 = a_3 = a_4$.



- **Straight flush**: Straight of the same suit. The cards are ordered as $a_1a_2a_3a_4a_5$ such that all the five cards have the same suit and that a_i is exactly one rank above a_{i+1} for all $1 \le i \le 4$. As a special case, if a_5 is Ace, a_4 can be 2. In this case, Ace is considered one rank below 2.



To compare two hands, first, we will compare the type of two hands. For example, when one hand is Four of a kind, and the other hand is Full house, Four of a kind always wins against Full house.

If the types of two hands are the same, we compare the ranks of the cards. We will order the cards as described above, and compare them one by one. Firstly, we will compare the first card. If a hand's first card has a higher rank, it wins. If the first cards of the two hands have the same rank, we will compare the second card, and so on. If the cards have the same rank in every position, no one wins. The suit of cards never matters in this comparison. For example, $\clubsuit 5, \diamondsuit 5, \heartsuit 5, \spadesuit 2, \clubsuit 2$ wins against $\diamondsuit 3, \spadesuit 3, \heartsuit 3, \diamondsuit A, \heartsuit A$: since they are both Full house, we will first compare the ranks of the three cards of a kind, and find that $5 > 3$.

Consider the case that the hole cards of Alice are $\clubsuit A, \diamondsuit 4$ and the hole cards of Bob are $\heartsuit 2, \spadesuit 3$. The community cards are $\spadesuit A, \heartsuit 4, \spadesuit 5, \clubsuit Q, \heartsuit Q$. The best hand for Alice (five cards among her hole cards and the community cards) is $\clubsuit A, \spadesuit A, \clubsuit Q, \heartsuit Q, \spadesuit 5$, which is Two pairs. The best hand for Bob is $\spadesuit 5, \heartsuit 4, \spadesuit 3, \heartsuit 2, \spadesuit A$, which is a Straight. Thus, Bob wins.

Players have many turns to make decisions. In each turn, the player has four choices: check, call, raise, or fold. These choices will be explained later.

The game between Alice and Bob proceeds in the following way:

1. **Stage 1**: The 52 cards are shuffled randomly.
   - The first two cards are put face-up, as Alice's hole cards.
   - The third and fourth cards are put face-up, as Bob's hole cards.
   - The fifth, sixth, and seventh cards are put face-down, as the flop.
   - The eighth card is put face-down, as the turn.
   - The ninth card is put face-down, as the river.

   Then, both Alice and Bob bet w dollars, and both have m dollars in hand.
2. **Stage 2**: A bet turn starts, where Alice and Bob bet in turns.
3. **Stage 3**: The flop is put face-up, then a bet turn starts.
4. **Stage 4**: The turn is put face-up, then a bet turn starts.
5. **Stage 5**: The river is put face-up, then a bet turn starts.
6. If no player chooses to fold in the above stages, their hands will be compared. Whoever has a better hand will be the winner. If the two hands are equal, the game will end with a draw.

When the game ends, suppose Alice bet a dollars and Bob bet b dollars. If the game ends with a draw, both players will receive $\frac{a+b}{2}$ dollars. If there is a winner, the winner will receive $a + b$ dollars and the loser will receive nothing.

If a player kept x dollars in hand at the end of the game, and received y dollars as the result of the game, then he or she will have a total of $x + y$ dollars when the game ends.

Each bet turn proceeds in the following way:

1. **Alice's turn**: Alice makes a decision.
2. **Bob's turn**: Bob makes a decision.
3. If Bob chooses to raise, then go back to Alice's turn. Otherwise, the bet turn ends.

Each player has four possible decisions. Suppose the current player bets a dollars and the other player bets b dollars. (One can prove that b is always no smaller than a.) Then:

- **check**. This choice is available only when $a = b$. If a player chooses to check, nothing will happen.
- **call**. This choice is available only when $a < b$. Suppose the current player has c dollars in the hand. When a player chooses to call, he or she has to further bet $b - a$ dollars.

- **raise**. This choice is always available. Suppose the current player has c dollars in the hand. Then, he or she can choose an integer $d \in (b, a+c]$, and further bet $d-a$ dollars.
- **fold**. This choice is always available. However, when a player chooses to fold, he or she will lose the game, and the game will end immediately.

Now, Stage 1 is finished: the four hole cards are put face-up, both players have bet w dollars, and both players have m dollars in hand. Suppose both players play optimally, wishing to maximize the expected amount of money they have when the game ends. Your task is to calculate this maximal expected amount of money for Alice.

Input

The first line contains an integer $T (1 \le T \le 10)$, the number of test cases.

For each test case, the first line contains two integers w and m $(0 \le w, m \le 50)$.

Then four lines follow. Each line contains two characters c and k separated by a space ($c \in \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}\}$, $k \in \{\texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{T}, \texttt{J}, \texttt{Q}, \texttt{K}, \texttt{A}\}$), where $c = \texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}$ represent clubs ($\clubsuit$), diamonds ($\diamondsuit$), hearts ($\heartsuit$), and spades ($\spadesuit$), respectively, $k = \texttt{T}$ represents the card 10, and the other values of k represent the respective card values.

The first two cards represent the hole cards of Alice, and the last two cards represent the hole cards of Bob. It is guaranteed that these four cards are pairwise distinct.

Output

Output a single line with a single irreducible fraction: the answer to the problem. For example, when the answer is 3.5, 0, 1, or 0.999, you need to output `7/2`, `0/1`, `1/1`, or `999/1000`, respectively.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>2
10 10
0 A
1 A
2 A
3 A
10 10
0 A
1 Q
2 K
2 J</pre></td><td><pre>20/1
557855/23782</pre></td></tr>
</table>