Problem A. Die

Source file name:	die.c, die.cpp, die.java, die.py
Input:	`Standard`
Output:	`Standard`

The boss of Binary Casino wants to have control over games played in his casino. This is especially true in the case of dice games, as from time to time people try to cheat their way to the victory by manipulating dice even after a throw has already occurred. Therefore, there is a camera installed to monitor the course of every single dice game. Dice recognition, however, is not an easy task and sometimes a camera may produce a faulty image which makes the task of recognition impossible.

There are rasterized data available from a camera which took pictures of several dice. The task is to write a program which determines the value of the top side of the die in the picture or detects that the picture is corrupted. Note that the picture is transformed in such way that the sides incident to the captured top side of the die are parallel to the x and y axis. However, it is unknown which of the four possible rotations of the top side is depicted.

Input

The input specifies the top side of a captured die. It consists of three lines, each with three characters "o" or ":" representing a dent or a smooth surface, respectively.

Output

Output a single line with either the number represented by the captured top side of the die or "`unknown`" if the image is incorrect.

Example

Input	Output
o:o :o: o:o	5
::o :o: o::	3
ooo ::: o:o	unknown

Problem B. Escalators

Source file name:	escalators.c, escalators.cpp, escalators.java, escalators.py
Input:	`Standard`
Output:	`Standard`

Binary Casino is a very special skyscraper building consisting of N floors connected by a tricky network of high speed escalators.

The floor connections are designed in a way that if there is an escalator going from floor A to floor B, then there is another escalator going from floor B to floor A as well. Also, for any two floors A and B, there is exactly one way to go from floor A to floor B.

Your manager decided to organize a promotion game to attract more customers. The game has the following rules:

- The game is played in one or more rounds.
- In each round a customer can choose a floor S on which that round starts. At this moment he earns some fixed number of tokens $t(S)$ associated with floor S. Then he may move to other floors using escalators.
- If a customer decides to take an escalator from floor A to floor B and has X tokens he has to pay $X - (X \text{ AND } t(B))$ tokens to enter floor B, where "AND" is a bitwise AND operator, "$-$" is a standard minus operator on numbers, and $t(B)$ is a number of tokens associated with floor B.
- A customer can decide to stop the round on any floor (including S) and keep the tokens from that round. Then he can start a new round from any floor if it is possible.
- No two rounds may have the same pair of starting and ending floors, not even in the opposite direction, i.e. when considering exchanged starting and ending floors.

Your manager is curious about the maximum number of tokens a customer can earn in the game.

Input

The first line of input contains an integer N ($1 \leq N \leq 3 \cdot 10^5$) describing the number of floors in the casino skyscraper. The second line contains N integers V_i ($0 \leq V_i < 2^{20}$). The $i-th$ integer V_i describes the number of tokens that a customer earns on the $i-th$ floor. After that, $N-1$ lines follow. Each line contains two integers A and B ($0 \leq A, B < N$) which describe an escalator connection between floors A and B.

Output

Output a single number - the maximum number of tokens a customer can earn.

Example

Input	Output
4 1 2 2 1 0 1 1 2 2 3	8
5 7 3 5 6 7 0 1 1 2 2 3 2 4	48

Problem C. Moving Furniture

Source file name:	furniture.c, furniture.cpp, furniture.java, furniture.py
Input:	`Standard`
Output:	`Standard`

Binary Casino had been open nonstop for a long period of time. It desperately needed renovation as the substantial damage to its carpet and furniture did not look good. While renovation took place, four-legged game tables from all game rooms were deposited in a storage room.

Your task is not difficult. You were asked by your boss to move the game tables to the exactly same place where they were positioned before. Each game table desk is a square and the table legs are located exactly under the corners of the desk. A good thing is that there are small hollows made by the table legs in the carpet at places where the game tables were standing before renovation and you know how many tables were in the room. Another good thing is that your boss does not remember where exactly were the tables placed. The bad thing is that your boss knows the total area of all game tables in the room and insists on the number being preserved.

You have to use every hollow in the carpet to place the game tables, placing one table leg into each hole. The tables cannot overlap.

Input

The first line of input contains an integer N ($1 \leq N \leq 3 \cdot 10^3$), the number of game tables. After that $4N$ lines follow, each containing two integers X and Y ($-10^9 \leq X, Y < 10^9$) which represent coordinates of a hollow in the carpet. The hollows are listed in an arbitrary order and no two hollows may have the same coordinates.

Output

Output a single number – the sum of areas of all game tables in the room. The output will be considered correct if it differs by at most 10^{-3} from the correct answer.

Example

Input	Output
4 1 0 1 -1 0 0 0 1 -2 -1 -2 1 -1 -2 -2 -3 -3 -2 -1 1 -1 -1 -3 -1 -3 1 0 -1 -2 3 0 3	11
1 0 0 3 4 -1 7 -4 3	25.0000

Problem D. Split Game

Source file name:	game.c, game.cpp, game.java, game.py
Input:	`Standard`
Output:	`Standard`

For a long time, rich clientele of Binary Casino has been requesting a new way to gamble their money. To fulfill their wishes, the director of Binary Casino decided to introduce a new game called Split Your Tokens.

This game is played only when a customer is about to exit the casino. Instead of exchanging tokens won during his visit, he may take up casino's challenge and bet all of his earned tokens on winning this game. Should the customer lose, all of his tokens are lost in favor of the casino.

When the game starts, the customer splits his tokens into N piles with not necessarily same amount of tokens in each pile. The customer and the casino then exchange turns – in this game we denote the customer as the first player and the casino as the second player. Each player in his turn decides which pile he wants to split and chooses a positive integer K which is smaller than the size of the selected pile. Then the player splits the selected pile into as many piles of size K as possible. If any tokens remain, they form another pile on their own. A player loses the game when he can not do any more splitting. The customer (first player) always plays first.

The director of Binary Casino is however not sure, whether this game will be profitable for the casino in the long term. Your task is thus to determine, for a given configuration of piles, which player wins when both players play optimally.

Input

The first line contains one integer N ($1 \leq N \leq 2000$), the number of piles. The second line contains a sequence of N integers P_i ($1 \leq Pi \leq 2000$), P_i represents the number of tokens in the i-th pile.

Output

Output a single line with either `First` or `Second`, depending on which player wins the game if both play optimally.

Example

Input	Output
3 1 2 3	First
3 1 2 2	Second

Problem E. Horsemeet

Source file name:	horsemeet.c, horsemeet.cpp, horsemeet.java, horsemeet.py
Input:	`Standard`
Output:	`Standard`

Traditional games such as chess or checkers, with slight modifications are also played in Binary Casino. However, not many people play them, as these games are often referred as boring. The visitors are more attracted to more dynamic games which cause adrenaline rushes. To attract players to traditional games, your boss wants to introduce a chess-based game called Horsemeet. The rules of the game are:

The game is played by two players on a 8×8 chessboard. One player plays a white knight and the other player plays a black knight. The players alternate in moves, the white knight moves first. In each move a knight is moved from its current position to a random valid position. Valid position within the chessboard is a position, which is two tiles away in one coordinate and one tile away in other coordinate from the original position. All moves to a valid position are equally probable. The first knight to move to a tile already occupied by the other knight wins.

In order to check whether this game could be at least partially interesting to visitors you have to determine the probability of victory for knights at given start positions. If the probabilities of victory for both knights differ by less than 10^{-6} the outcome of such configuration is a draw.

Input

The first line of input contains two integers A and B ($1 \leq A, B \leq 8$), the start position of the white knight. The second line of each input consists of two integers C and D ($1 \leq C, D \leq 8$), the start position of the black knight. You can assume both positions are distinct.

Output

Output the knight with a higher probability of victory: `white` or `black`. In case of equal probabilities output `draw`.

Example

Input	Output
1 1 4 7	white
1 1 8 8	black

Problem F. Lighting

Source file name:	lighting.c, lighting.cpp, lighting.java, lighting.py
Input:	`Standard`
Output:	`Standard`

The lighting system in Binary Casino is controlled by a very complex and secure mechanism, which is connected to a central control console. At the console, the state of each light is stored as one bit of information (0 = the corresponding light is off, 1 = light is on), so the complete state of all lights in the building may be represented by a binary number a.

To avoid manipulation by unauthorized persons, the lighting system has a special method to control the lights. Should one want to change the configuration of the lights, it is necessary to enter a binary number b which gets added to the original configuration a using standard integer summation.

You need a particular number of lights to be switched ON and you are curious what are your chances of success. How many suitable binary numbers are there?

Input

The first line of input contains two integers N and K ($1 \leq N \leq 1000, \quad 0 \leq K \leq N$), N representing the number of bits of a and of b, and K the target number of lights to be switched ON. The second line contains a binary integer a of length N.

Output

Print the number of different nonnegative N-bit integers b such that the sum $a + b$ has exactly K bits set to 1. As the result might be large, output it modulo $10^9 + 7$.

Example

Input	Output
4 2 1100	5
10 5 1000100111	260
13 1 0000000000000	13

Problem G. Locker Room

Source file name:	lockers.c, lockers.cpp, lockers.java, lockers.py
Input:	Standard
Output:	Standard

There are several strange rooms in Binary Casino and one of them is a locker room. You have to enter the locker room several times a day, two times being a minimum (before and after your shift). There is no key or access code needed to enter the locker room. There is a brand new security lock system instead.

The lock system presents you with a generated puzzle which you have to solve every time you want to enter the locker room. This system prevents possible intruders to enter the locker room, as the puzzle takes them a long time to solve. Only employees, after working in the casino for some time, manage to master the puzzle.

It is your second week in the casino and you have already been late three times because you didn't manage to solve the puzzle quickly enough. You therefore decided to write a program which solves the puzzle. The puzzle is as follows:

You are given a cyclic string of N lowercase english letters. You have to choose and mark substrings (continuous segments of characters) of a given length K until each character in the string is marked. Marking a substring does not change the original string and each character can be marked multiple times. The task is to print the lexicographically maximal substring among chosen substrings. In addition, the printed substring has to be lexicographically minimal possible.

For example, let "`acdb`" be the given string of length $N = 4$ and let $K = 3$. Then you can choose substrings "`acd`" and "`bac`" to mark the whole string. The puzzle solution is "`bac`".

Input

The first line of input contains two integers N and K ($1 \leq N \leq 5 \cdot 10^5$, $1 \leq K \leq N$), describing the length of the given string and the length of marked substrings. The second line contains N lowercase english letters – the given cyclic string.

Output

Output the lexicographically maximal substring among chosen substrings under the condition the result is lexicographically minimal possible.

Example

Input	Output
4 3 acdb	bac
6 2 aababa	ab
10 4 abaaabaaba	aaba

Problem H. Numbers Generator

Source file name:	numbers.c, numbers.cpp, numbers.java, numbers.py
Input:	Standard
Output:	Standard

Among the most popular games in Binary casino is a game called "The Binary Generator". It is played by multiple players and with a single coin. Each player first chooses a sequence of heads and tails of a given length. After that, players or the casino start flipping the coin and the winner is the player whose sequence first appears as a contiguous subsequence of the flip results.

You believe all sequences chosen by players are equally good and so the choice of the sequence does not matter. However, after losing all of your money, you became somewhat doubtful of that. Write a program to prove you wrong. For a given list of sequences of heads and tails of the same length, write the expected number of coin flips which have to be performed until one of the players' chosen sequences appears as a contiguous subsequence in the flip sequence. The expected number of coin flips is the average length of a flip sequence over all possible flip sequences resulting in some player's victory, where each flip sequence is weighted by its probability.

Input

The first line of input contains two integers W and B ($1 \leq W \leq 10$, $1 \leq B \leq 30$), the number of players' sequences and the length of players' sequences, respectively. Next, W lines follow, each consisting of a sequence of B letters. Each letter is either "H" for heads or "T" for tails.

Output

Output a single number – the expected number of flips. The output will be considered correct if it differs by at most 0.1 from the correct answer.

Example

Input	Output
1 1 H	2.0
2 3 HHT THT	5.0
2 3 HHH HHT	7.0

Problem I. Rullete

Source file name:	rullete.c, rullete.cpp, rullete.java, rullete.py
Input:	`Standard`
Output:	`Standard`

Binary Casino has established a new department to attract families with children. One of its first tasks is to design a game for children which will be not so difficult to play. The result of a week of hard work is a game called Rullete (sic!).

Each player gets a hand of 5 cards. Cards in the hand are ordered as first, second, . . ., fifth. Each card is described by a rank and a suit. Card's rank can be one of 2, 3, ..., 10, J, Q, K, A and its suit can be one of D, H, C, or S (Diamonds, Hearts, Clubs, or Spades). Cards 2–10 have scores equal to their rank and cards J, Q, K, and A all have a score of 10. Initially, a player's hand has a value given by the sum of scores of the cards in his hand. The initial hand value is then changed according to the rules of the game. To make croupier's life easier, your task is to calculate the final value of you hand of cards after all of the following fourteen rules are applied in the given order:

1. If you have at least 4 cards in your hand then add 1 to the value. Also add to the value the product of the number of J's in your hand and the score of the first card in your hand.
2. If you have at least 2 cards of the same suit in your hand then multiply the value by 2.
3. If you have at least one card of each suit in your hand then multiply the value by 2.
4. If the count of black (Clubs and Spades) and the count of red (Hearts and Diamonds) cards in your hand differ then add the absolute value of the difference of the counts to the value.
5. If the value is currently even then add all positive integer divisors of the value (including 1 and the value itself) to the value.
6. If there are exactly 4 cards of rank 7 in your hand then subtract 11 2 from the value.
7. If the value is currently non-negative then add the score of the lowest score card in your hand to the value.
8. If the value is currently negative then multiply the value by -1.
9. If there are at least 3 cards of Diamond suit in your hand then add 1 to the value and then simultaneously swap ranks of all 6's to 9's, all 9's to 6's, all 2's to 5's, and all 5's to 2's in your hand.
10. If there is a **straight** in your hand then add five times the number of A's in your hand to the value.
11. If the value was modified by more than 8 rules so far then add the number of 1's bits in the binary representation of the value to the value.
12. If there is at least one card of rank 2 in your hand then apply once again the last rule which changed the value (after that continue with rule 13).
13. If there is at least one card of rank 2 in your hand then add the product of all distinct superfactors of the value to the value. A superfactor is a prime raised to the highest integer power so that it divides the value evenly.
14. If the value is 674 you win!

A **straight** is a set of any 5 consecutive cards in the following order: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A.

Input

Input consists of a single line with five space-separated card descriptions, each containing the card rank immediately followed by its suit.

Output

Output the score after applying all of the rules.

Example

Input	Output
2D 5D JD KC AC	170
QC 8D JD 10S 9D	126

Problem J. Security Guards

Source file name:	security.c, security.cpp, security.java, security.py
Input:	`Standard`
Output:	`Standard`

In the course of the last few weeks, Binary Casino has been afflicted by a local crime wave which is primarily focused on casinos in the neighborhood. Although there are surveillance cameras installed in Binary Casino, thieves usually manage to sneak out with relative ease as there is almost nobody patrolling in the casino.

After the theft of all Friday's earnings, the manager of Binary Casino has lost his patience and decided to reinforce security of the casino by hiring a vast number of security guards. However, nobody in the casino was capable of coming up with a plan on how to distribute guards across the whole casino to maximize security. Security guards are thus scaterred across the casino in no systematic way. Fortunately, their locations can be described by integer coordinates in a 2D plane.

Because of the uneven distribution of security guards, in case of a reported robbery it is very hard for security supervisors to determine which guard is closest to the location of the incident. The task is even harder due to the constrained space in the casino which consists of endless aisles of slot machines. This limitation forces each guard to travel from one location to another in a sequence of steps. In each step, he/she can change each of his/her coordinates by 1, 0 or -1. The distance between two locations is equal to the minimum number of steps the guard has to do to get from one location to the other one.

The task is, for a given locations of guards and a set of locations of security incidents, to compute for each incident its smallest distance to any of the guards. This will allow security supervisors to alert appropriate guards and will greatly increase the casino security.

Input

The first line of input contains two integers N and Q ($1 \leq N, Q \leq 3 \cdot 10^5$), the number of guards and the number of security incidents, respectively. After that, N lines follow. Each of these lines contains two integers X and Y ($0 \leq X, Y \leq 5000$) which describe coordinates of a guard in a 2D plane. Next, Q lines follow. Each of these lines contains two integers A and B ($0 \leq A, B \leq 5000$) which describe coordinates of a security incident.

Output

For each of Q security incidents output a line containing shortest distance to any of the security guards, measured in steps.

Example

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