Blowfish

Time limit: 4 seconds

Print the figure exactly as it appears.

Input

None

Output

See below. Art by The Kat.

```
                         .
                          A       ;
                |   ,--,-/ \---,-/|  ,
               _|\,'. /|      /|   `/|-.
           \`.'    /|      ,            `;.
          ,'\   A     A         A   A _ /| `.;
        ,/  _              A       _  / _   /|  ;
       /\  / \   ,  ,           A  /    /     `/|
      /_| | _ \         ,     ,             ,/  \
     // | |/ `.\  ,-      ,       ,   ,/ ,/      \/
     / @| |@  / /'   \  \      ,              >  /|    ,--.
    |\_/   \_/ /      |  |           ,  ,/        \  ./' __:..
    |  __ __  |       |  | .--.  ,         >  >   |-'   /     `
  ,/| /  '  \ |       |  |     \      ,           |    /
 /  |<--.__,->|       |  | .    `.        >  >    /   (
/_,' \\  ^  /  \     /  /   `.    >--            /^\   |
      \\___/    \   /  /      \__'     \   \   \/   \  |
       `.   |/          ,  ,                  /`\    \  )
         \  '  |/    ,       V    \          /        `-\
          `|/  '  V      V           \    \.'            \_
           '`-.       V       V        \./'\
               `|/-.      \ /   \ /,---`\
                /   `._____V_____V'
                           '     '
```

She Sells Sea Shells

Time limit: 4 seconds

Sandy enjoys selling the shells that she finds on the ocean floor to locals. Every morning, she goes out to find new shells to sell at her store.

If Sandy does not find an odd number of blue shells and an even number of green shells, she takes this as a bad omen and closes her shop for the day. She's also been known to close her shop when she doesn't find at least 10 shells in total. Knowing this, help the residents of Swimsuit Land predict whether or not Sandy's Sea Shell Shop will open for the day, based on her haul.

Input

The first line of the input file contains an integer ***T*** ($1 \le T \le 1000$) indicating the number of cases that follow. Each of the next ***T*** lines two positive integers, ***M*** and ***N***, with ***M*** indicating the number of blue shells and ***N*** indicating the number of green shells ($1 \le M, N \le 10^9$).

Output

For each test case, print "OPEN" if Sandy's Sea Shell Shop will be open for the day. Otherwise, print "CLOSED".

Sample Input 0	Sample Output 0
3 23 12 1 7 3 4	OPEN CLOSED CLOSED

Sample Input 1	Sample Output 1
10 5 7 4 13 42 58 1 53 52 23 23 2 5 4 21 32 34 21 1 54	CLOSED CLOSED CLOSED CLOSED CLOSED OPEN CLOSED OPEN CLOSED OPEN

Cryptopwatty

Time limit: 4 seconds

Plankton has been trying to steal the recipes of Krusty Krab's Krabby Patty for years and almost succeeded, but this time, Mr. Krabs has used a *simple substitution cipher* to encrypt his recipe.

First, the recipe is stripped of any punctuation, and all letters are capitalized. After this, it consists of several lines of text, which only contain uppercase letters (A-Z) and spaces.

To encrypt the message, a secret key is used, which is a string of 26 distinct English letters (for example, "`DGJHILKMANOQRSZCXBUFWPEVYT`"). This represents a substitution, where each letter of the alphabet is changed to the corresponding letter in the key. In particular, the ***k***-th letter of the alphabet maps to the ***k***-th character of the key. For our example key, we would have: A → D, B → G, C → J, …, Z → T.

Mr. Krabs has already applied this substitution to all the letters in the recipe (spaces are not changed). Given the substitution key and the encrypted recipe, your task is to undo the encryption and determine the original text.

Input

The first line contains ***N***, the number of lines of encrypted text (1 ≤ ***N*** ≤ 100). The next line contains the encryption key, a string of 26 unique uppercase letters. The next ***N*** lines provide the encrypted message, which may have spaces and contains at most 1000 characters in total.

Output

Using the provided key, print the plaintext message.

Sample Input 0	Sample Output 0
2 EFGHIJKLMNOPQRSTUVWXYZABCD OAEFFC TAEXXC WEPX	KWABBY PWATTY SALT

Sample Input 1	Sample Output 1
1 DGJHILKMANOQRSZCXBUFWPEVYT CEDSOFZS AU D RIRI	PWANKTON IS A MEME

Oh, No, My Letters!

Time limit: 4 seconds

A waternado ripped through all of Mr. Krabs's old, old telegrams! With the help of SpongieBoi and Patricia Decagon, he's been able to (somewhat) recreate the messages with the pieces left behind. Each piece contains either one or two letters. Given the recreated message and the original pieces, verify if the recreated message is valid. Remember, the waternado may have swept away some parts of the telegram. The recreated message is valid only if it contains all the pieces of the original telegram as non-overlapping substrings.

Input

The input will contain 2 lines of input, formatted as follows:

- One line with the integer M ($1 \le M \le 10$), followed by M unique, space-separated snippets of alphanumeric text, indicating the pieces of the telegram. It is guaranteed that each of the smaller pieces will occur in the recreated message exactly once.
- One line containing the recreated message, with a maximum length of 100 characters.

Output

Print "Potentially!" if the recreated message is valid, given the remaining fragments of the original message. Otherwise, print "No, No, No!!!".

Sample Input 0	**Sample Output 0**
`3 YO AR M` `YOU MARKED IT`	`Potentially!`

Sample Input 1	**Sample Output 1**
`5 AR A ER TH OU` `ARE YOU THERE`	`No, No, No!!!`

Krabby Land

Time limit: 4 seconds

A group of Bikini Bottom residents is planning a trip to the Krabby Land amusement park (the one from season 3, episode 57a). However, all attendees must be either above a certain age or accompanied by someone who is above the required age. In addition, each adult sea creature can accompany at most two under-aged creatures (i.e., there must be at least one adult for every two kids). Unfortunately, at present, not enough members of the group are old enough. Your task is to determine how long the group must wait before attending Krabby Land.

Input

The first line of input contains three integers, the first, P ($1 \le P \le 100$), denoting the number of people in the group and the second and third, A_y and A_d ($0 \le A_y, A_d < 365$), indicating the minimum age for attendees considered adults in that group, in years and days. The following P lines contain information about the group members, with each line containing an alphabetic string N ($1 \le |N| \le 100$), representing the creature's name and two integers, Y and D ($0 \le Y, D < 365$), denoting the number of years and days in the creature's age, respectively.

Creatures' names will not contain spaces.

Output

Display the amount of time needed until a sufficient number of sea creatures in the group are at or above the required age. Output should be in the format "XX year(s) and ZZ day(s)". Assume that each year has exactly 365 days.

Sample Input 0	Sample Output 0

`2 21 0`
`Sandy 18 59`
`Pearl 16 224`
	`2 year(s) and 306 day(s)`

Sample Input 1	Sample Output 1

`5 24 364`
`Larry 23 190`
`Squilliam 17 34`
`Fred 16 142`
`Nat 14 29`
`Jenkins 17 315`
	`7 year(s) and 49 day(s)`

Krusty Krab Orders

Time limit: 4 seconds

Customers from all over Bikini Bottom flock to the Krusty Krab each day. Three main items are available—krabby patties, coral bits, and kelp shakes. Anyone who orders a krabby patty also orders either a coral bits or a kelp shake to go with it and nothing else, coral bits are never ordered on their own, and no item is ordered twice by the same customer.

In summary, each customer makes only one order out of the three possibilities (either the krabby patty/coral bits combo, the krabby patty/kelp shake combo, or just a kelp shake). Considering that krabby patties cost \$1.25, coral bits cost \$1.00, and kelp shakes cost \$2.00, your task is to determine how many of each item were sold on a particular day.

Input

The first line of input contains one integer, D ($1 \le D \le 100$), denoting the number of days to follow. The following D lines contain information about each day, with each line containing an integer, I ($1 \le I \le 2000$), representing the total number of items sold that day, a floating-point number, S ($1.00 \le S \le 10000.00$), denoting the total amount of money earned from sales that day in units of dollars, and an integer, C ($1 \le C \le 1000$), representing the total number of customers for the day.

Output

For each day, display the quantity of each item sold. Output should be in the format "XX krabby patties, YY coral bits, ZZ kelp shakes".

Sample Input 0	**Sample Output 0**

`2`
`8 12.75 5`
`84 133.25 59`
	`3 krabby patties, 1 coral bits, 4 kelp shakes`
	`25 krabby patties, 16 coral bits, 43 kelp shakes`

Sample Input 1	Sample Output 1
3 173 203.00 89 190 269.25 125 228 345.25 123	84 krabby patties, 80 coral bits, 9 kelp shakes 65 krabby patties, 62 coral bits, 63 kelp shakes 105 krabby patties, 32 coral bits, 91 kelp shakes

Magic Jellyfish Field

Time limit: 4 seconds

It's jellyfish catching season, and this year, Spongebob and his friends are planning something new: an expedition to a magic field that attracts jellyfish! However, demand is high, so they must choose a rectangular region of the grid to reserve in advance.

The field is in the shape of an ***R*** × ***C*** rectangular grid of locations ($1 \le R*C \le 1000$). Each of the locations attracts a specific number of jellyfish, which may be different for different locations. Spongebob's plan is to choose and rent some rectangular region of the grid, with positive area and sides parallel to the horizontal and vertical directions. He will invite one friend to help him catch jellyfish for each distinct location within the rectangle. However, Spongebob does not want any of his friends to feel left out, so the optimal region should maximize the average number of jellyfish caught by each of his friends.

In other words, the goal is to pick a region that maximizes the total number of jellyfish caught, divided by the area of the region. Your task is to compute this maximum value.

Input

The first line of text contains two integers, ***R*** and ***C***. The next ***R*** lines each contain ***C*** space-separated integers (each between 0 and 10^6), describing the number of jellyfish that can be caught at each field location.

Output

Print the maximum value for the average number of jellyfish among all locations in a rectangular subregion of the field, rounded down to the nearest integer if fractional.

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

Sample Input 1	Sample Output 1
4 4 1 2 3 4 5 100 100 6 7 100 100 8 9 10 11 12	100

Plankton Chaser

Time limit: 4 seconds

Plankton has just escaped the Krusty Krab with the secret formula! In his attempt to get away, Plankton begins running due east away from the Krusty Krab at time t = 0. However, it isn't long until Mr. Krabs notices his precious formula is missing and sends Spongebob out to chase Plankton down. As soon as Spongebob leaves the Krusty Krab, Plankton turns and begins heading due north. Assuming Spongebob takes the optimal route, your task is to determine how long it will take him to reach Plankton and recover the secret formula.

Input

The first line of input contains one integer, N ($1 \le N \le 100$), denoting the number of test cases to follow. The following N lines each contain a floating-point value, t_s ($0 \le t_s \le 10000$), representing the time in seconds after Plankton escapes that Spongebob begins to chase after him, and two more floating-point values, v_p and v_s ($2 \le v_p + 1 \le v_s \le 10000$), representing the speeds in sea-meters per second of Plankton and Spongebob, respectively.

Output

For each case, display the time in seconds after t = 0 when Spongebob reaches Plankton. Your solution will be judged correct if it is within 10^{-6} absolute error of the correct answer.

Sample Input 0	Sample Output 0
3 2.2 5 13 1.875 8 17 1.944 18 55.5	3.1166667 2.8750000 2.6105143

Sample Input 1	Sample Output 1
5 0 200 666 10000 1 2 10000 1 10000 632 5000 10000 .01 50 300	0.0000000 15773.5026919 10001.0000000 996.8853701 0.0116903

Grid Arrangements

Time limit: 4 seconds

SpongeBob is making Krabby Patties at the Krusty Krab. Krabby Patty orders come in groups of size ***N***. Each time SpongeBob receives a new order of ***N*** Krabby Patties, he cooks all ***N*** Krabby Patties on his stove simultaneously. However, since SpongeBob prefers to be organized, he insists that the ***N*** Krabby Patties must be in the shape of a completely filled rectangular grid. SpongeBob wonders how many ways this is possible in, so for each ***N***, he lets ***f(N)*** be the number of ways to arrange the ***N*** Krabby Patties in a completely filled rectangular grid (assume SpongeBob's stove is infinitely long and wide). For example, if ***N*** = 4, then **f(*N*)** = 3, because SpongeBob can arrange his Krabby Patties in a 1x4 grid, a 2x2 grid, or a 4x1 grid, and the 1x4 and 4x1 grids are counted separately.

Since SpongeBob enjoys adding lots of numbers together, he takes this definition a step further and wonders what the value of ***f(1) + f(2) + ... + f(N)*** is. For each given ***N***, help SpongeBob compute this sum.

Input

The first line of the input file contains an integer ***M***, the number of integers that follow (1 ≤ ***M*** ≤ 100). Each of the following ***M*** lines contains an integer ***N*** (1 ≤ ***N*** ≤ 10^5).

Output

For each value of ***N***, print the value of ***f(1) + f(2) + ... + f(N)***, with each sum on a new line.

Sample Input 0	Sample Output 0
2 3 4	5 8

Explanation: Since ***f(1) = 1***, ***f(2) = 2***, ***f(3) = 2***, ***f(4) = 3***, for the first sum we have ***f(1) + f(2) + f(3) = 5***, and for the second sum we have ***f(1) + f(2) + f(3) + f(4) = 8***.

You Snooze, You Lose

Time limit: 4 seconds

Ahhhh! SpongeBob snoozed the alarm again and has only 15 minutes to get to work at the Krusty Krab. Spongebob's neighborhood is a 10-by-10 rectangular grid, where each square is either a paved road, which he can walk on, or a wall, which he cannot pass through. Each minute, Spongebob can move from his current position to one adjacent square in any of the four orthogonal directions (left, right, down, up). He always takes the shortest path to his destination.

Help Spongebob determine whether he will make it to work on time!

Input

The input will consist of ten lines of ten characters each, representing a map of the land between SpongeBob's house and the Krusty Krab. The character "S" is located at the starting position, and "K" is located at the Krusty Krab. Also, the characters "#" denote walls, and "*" denote paved roads that SpongeBob can move through.

There is guaranteed to be at least one path from home to work.

Output

Determine whether or not SpongeBob will arrive on time to work. If so, output "On time: XX minutes", replacing XX with the minimum amount of time it will take him to reach the Krusty Krab. If not, output "Late: XX minutes", where XX is the number of minutes by which SpongeBob will be late (in excess of 15 minutes).

Sample Input 0	Sample Output 0

`#********#`
`#S##***#*#`
`#****#****`
`#*#####*##*`
`#****##K##*`
`**#*##*###`
`*##******#`
`#######*##`
`######**##`
`##########`
	`On time: 10 minutes`

Sample Input 1	Sample Output 1
########## #S##***#*# #****#***# #*######## #***##*### #*#*##*### ###******# #######*## ##K*****## ##########	Late: 3 minutes

Prime Establishment

Time limit: 4 seconds

Mr. Krabs wants to win the award for top Prime Eating Establishment in Bikini Bottom. However, in order to win, the Krusty Krab must have a prime number of customers every day. Additionally, the award requires that the chosen establishment be constantly improving. As such, the number of customers each day must be greater than the day before.

Your task is to determine the minimum number of customers the Krusty Krab must have on a particular day to be eligible to win the Prime Eating Establishment award.

Input

The first line of input contains one integer, ***D*** ($1 \leq D \leq 100$), denoting the number of days that follow. The following ***D*** lines each contain a prime integer, ***C*** ($1 \leq C \leq 10000$), representing the total number of customers that day. Each day is independent of all other days.

Output

For each day, display the minimum number of customers the Krusty Krab must have on the following day.

Sample Input 0	Sample Output 0
2 53 7	59 11

Sample Input 1	Sample Output 1
12 2 71 359 997 2003 6863 9901 9931 9941 9949 9967 9973	3 73 367 1009 2011 6869 9907 9941 9949 9967 9973 10007

Cheap Krabs

Time limit: 4 seconds

You are given an array $\boldsymbol{A}$ of $\boldsymbol{N}$ positive integers ($1 \le \boldsymbol{N} \le 10^5$), and are interested in its *subarrays*, i.e., sequences of consecutive elements within the array. The subarray from $\boldsymbol{i}$ .. $\boldsymbol{j}$ inclusive (where $\boldsymbol{i} \le \boldsymbol{j}$) is given by the sequence of numbers { $\boldsymbol{A_i}$ $\boldsymbol{A_{i+1}}$ $\boldsymbol{A_{i+2}}$ … $\boldsymbol{A_j}$ }.

Define the *cheap-score* of a subarray to be its length, minus the absolute differences between pairs of consecutive elements. For example, if the given array is { 1, 2, 5, 7, 6, 7, 7, 5, 6, 3 }, then one possible subarray would be { 6, 7, 7, 5, 6 }. The cheap-score of this subarray is

$$length - differences\ = 5 - |6-7|\ - |7-7| - |7-5| - |5-6| = 5 - 1 - 0 - 2 - 1 = 1.$$

Your task is to determine the maximum cheap-score of any subarray of the given array.

Input

The first line contains $\boldsymbol{N}$, the length of the array. The following line contains $\boldsymbol{N}$ space-separated integers $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, …, $\boldsymbol{A_N}$, the elements of the array ($1 \le \boldsymbol{A_j} \le 1000$).

Output

Print the maximum cheap-score.

Sample Input 0	**Sample Output 0**
5 1 1 1 1 9	4

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

Explanation: For Sample Input 0, the optimal subarray is the first four elements { 1 1 1 1 }, which has cheap-score equal to 4 - 0 - 0 - 0 = 4. For Sample Input 1, the optimal subarray is the entire array, which has a cheap-score of 9 - 0 - 0 - 0 - 2 - 1 - 1 - 0 - 0 = 5.

Primes Are Hard

Time limit: 4 seconds

Uh oh! SpongieBoi has a math assignment due in an hour, but he's not very good at prime factorization. He's confident in his multiplication, but he's never sure if a number is prime. He's tried his best, but he needs someone to check his answers. Help SpongieBoi check his prime factorization!

Input

The first line of the input file contains an integer T, with $1 \le T \le 100$, denoting the number of SpongeBob's equations that follow, all of which will be mathematically correct. Each equation will then be given in the format $N = P_1 \times P_2 \times P_3 \times \ldots \times P_k$, where:

- N is the product ($2 \le N \le 10^{12}$)
- The number of factors, k, satisfies $1 \le k \le 40$
- Each of the P_j, in increasing order, represent SpongeBob's factorization. ($2 \le P_j \le 10^6$)

Output

If SpongeBob's answer was incorrect, your program should output the each following sections on separate lines:

- "WRONG!"
- For each unique incorrect "prime" that SpongeBob had in his answer, print "XX is not a prime number." on separate lines. Even if the same incorrect "prime" appears more than once in an equation, only print this statement once.
  - The incorrect "primes" should be output in increasing order.
- The correct prime factorization, with factors in increasing order.

Otherwise, if SpongeBob's factorization is correct, your output should consist of two lines:

- "correct!"
- The correct equation from the input, repeated.

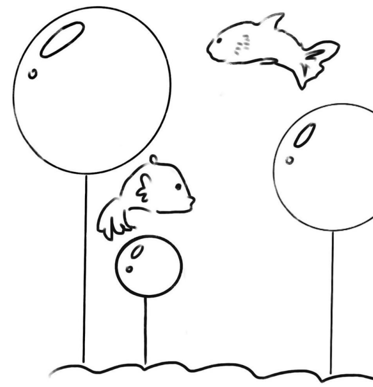
Sample Input 0	Sample Output 0
3 6 = 6 15 = 3 x 5 21 = 3 x 7	WRONG! 6 is not a prime number. 6 = 2 x 3 correct! 15 = 3 x 5 correct! 21 = 3 x 7

Sample Input 1	Sample Output 1
8 3 = 3 9 = 3 x 3 188 = 4 x 47 17901 = 3 x 27 x 221 248 = 2 x 2 x 2 x 31 44051 = 7 x 31 x 203 193027 = 43 x 67 x 67 3233 = 53 x 61	correct! 3 = 3 correct! 9 = 3 x 3 WRONG! 4 is not a prime number. 188 = 2 x 2 x 47 WRONG! 27 is not a prime number. 221 is not a prime number. 17901 = 3 x 3 x 3 x 3 x 13 x 17 correct! 248 = 2 x 2 x 2 x 31 WRONG! 203 is not a prime number. 44051 = 7 x 7 x 29 x 31 correct! 193027 = 43 x 67 x 67 correct! 3233 = 53 x 61

Angry Fish

Time limit: 4 seconds

The people of Bikini Bottom are mad at Squidward for practicing too loudly! As a result, N fish have come to protest at Squidward's house by knocking angrily on his front door. Assume that the j-th fish arrives at Squidward's doorstep at the start of day B_j and leaves Squidward's doorstep immediately before the start of day C_j, where $1 \le B_j < C_j \le 10^9$.



However, to complicate things further, there are 20 different species of fish in Bikini Bottom, numbered from 0 to 19. If there are an even number of fish of a given species in front of Squidward's house on some day, those fish will pair up and distract each other, so none of those fish will actually knock on Squidward's door. Meanwhile, if the number of fish of a given species is odd, then the fish will not be distracted, and all of them will knock on Squidward's door.

Your task is to determine the number of days for which at least one fish knocks on Squidward's door, given the arrival day, departure day, and species number of each fish.

Input

The first line of the input file contains N, the number of fish ($1 \le N \le 10^5$). The j-th of the following N lines contains three spaced integers A_j, B_j, and C_j, respectively denoting the species number, the arrival day, and the day on which the fish leaves. It is given that $0 \le A_j \le 19$.

Output

Output the total number of days for which at least one fish knocks on Squidward's door.

Sample Input 0	Sample Output 0
3 1 2 4 2 10 100 1 3 4	91

Explanation: On day 2, the first fish knocks on Squidward's door. On day 3, the first and third fish distract each other and nobody knocks on Squidward's door, and at the end of day 3, both fish leave. Then the second fish knocks on Squidward's door on days 10, 11, 12, …, 99, so in total there are 91 days of knocking.

Tree Labels

Time limit: 4 seconds

You are given a tree with N vertices ($1 \le N \le 2 \times 10^5$) indexed 1, 2, …, N, with positive integer label A_i ($1 \le A_i \le N$) on vertex i.

For each vertex v in the tree, your task is to determine the number of distinct labels on the path from vertex 1 to v.

Input

The first line of input contains N, the number of vertices in the tree.

The second line of input contains N integers A_1, A_2, …, A_N, the labels on successive vertices of the tree. These may not necessarily be distinct.

The next N-1 lines of input each contain two integers u, v, representing an undirected edge between vertices u and v. It is guaranteed that these edges will form a tree.

Output

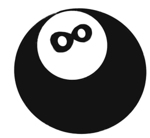
Display N lines, where line v contains the number of distinct labels on the path described above.

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

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

Pool Table

Time limit: 4 seconds



You are playing a game of pool on an irregular pool table with ***T*** sides ($3 \leq T \leq 20$). The table is a convex simple polygon, defined with ***T*** vertices given in order, where the border of the table is produced by connecting V_1 to V_2, V_2 to V_3, and so on.

There are two small balls within the interior of the polygon, at different locations. Given the location of the starting ball and a target ball, calculate the counterclockwise angle in degrees, from the positive-x direction, which the ball must be fired at to strike all ***T*** sides of the pool table, in order, starting from the first side (V_1--V_2), and hitting the target ball at the very end. It is guaranteed that there exists a unique solution for all the given test data.

Input

The first line of input contains one integer ***T***, the number of vertices the table has. The ***j***-th of the next ***T*** lines each contain two integers X_j and Y_j, giving the coordinates of the vertices in order ($|X_j|, |Y_j| \leq 10^4$). The next line contains the location of the target ball, and the last line of input contains the location of the white ball, both given as pairs of integer coordinates.

Output

A single line displaying the angle which the white ball needs to be hit at. Any solution within 10^{-6} absolute error of the actual answer will be judged correct.

Sample Input 0	**Sample Output 0**
4 0 0 0 4 4 4 4 0 1 2 1 3	138.81407483

Sample Input 1	**Sample Output 1**
3 0 0 5 -1 2 3 2 0 1 1	274.10419353

Lucky Hands

Time limit: 4 seconds

SpongeBob is going to play a game of cards with Patrick Star, but everyone knows that Patrick is an expert in this game. If SpongeBob wants even a chance at winning, he knows that his cards must be in his favor. Help SpongeBob determine his luck when drawing cards.

In this game, each player draws 10 cards, and the more special combinations that a player has, the better. Cards are represented by their ranks (2 - 10, J - K, A) and their suit (clubs, diamonds, hearts, spades). Below is a list of potential combinations.

Value	A	2	3	4	5	6	7	8	9	10	J	Q	K
Rank	1	2	3	4	5	6	7	8	9	10	11	12	13

Combinations (listed in order of decreasing importance)		**Example**
Straight Flush**	Five cards of sequential rank, all of same suit.	2C, 3C, 4C, 5C, 6C
Four of a Kind	Four cards of same rank (all suits must be represented).	7C, 7D, 7H, 7S
Full House	Three cards of one rank, 2 of another.	9C, 9D, 9H, 2C, 2S
Straight**	Five cards of sequential order, not all of same suit.	9S, 10H, JC, QD, KD
Three of a Kind	Three cards of same rank.	3C, 3D, 3H
Pair	Two cards of same rank.	JH, JC
**straights cannot “wrap around” → Q,K,A,2,3 is not considered a straight		

From a set of hands, determine the number of each combination in each hand, as well as the average number of combinations from all the hands.

Input

The first line of the input file contains an integer ***T*** (1 ≤ ***T*** ≤ 50), indicating the number of hands given. The ***k***-th of the next ***T*** lines contains a string that represents the ***k***-th hand, with 10 cards separated by commas. There will not be duplicate cards in each hand.

Output

For each hand, you should output the following on separate lines:

- “Hand #XX”

- For each combination, print the number of times the combination occurs in the hand given in the format “<COMBINATION NAME>: XX”.
    - Combinations should be printed in the order of greatest to least importance.
    - Since straight flushes are worth the most points in the game, SpongeBob wants to know exactly what his straight flushes are. If the hand contains a straight flush, please print the flush with the cards appearing in the order that they were given. If more than one straight flush exists, the order of the flushes should follow alphanumeric order. Each flush should be printed on separate lines.
- Output an empty line between each pair of separate hands.

Following the information about each hand, print the following on separate lines:
- “Average”
- For each combination, print the average number of times that combination appeared in the hands that SpongeBob drew, rounded to the nearest hundredth place in the format “<COMBINATION NAME>: XX”.
    - Combinations should be printed in the order of greatest to least importance.

<table>
<tr><th>Sample Input 0</th><th>Sample Output 0</th></tr>
<tr><td><pre>2
2C,8C,JC,10C,3S,5S,9C,KC,QC,2H
3H,7D,3C,5D,6H,8D,4D,10H,JH,AC</pre></td><td><pre>Hand #1
Straight Flush: 2
8C,JC,10C,9C,QC
JC,10C,9C,KC,QC
Four of a Kind: 0
Full House: 0
Straight: 0
Three of a Kind: 0
Pair: 1

Hand #2
Straight Flush: 0
Four of a Kind: 0
Full House: 0
Straight: 3
Three of a Kind: 0
Pair: 1

Average
Straight Flush: 1.00
Four of a Kind: 0.00
Full House: 0.00
Straight: 1.50
Three of a Kind: 0.00
Pair: 1.00</pre></td></tr>
</table>

Sample Input 1	**Sample Output 1**
3 9H,8D,9D,KD,9S,10H,2C,KH,2D,3D 6D,KS,7D,8D,AH,6C,3D,5D,4D,AS 4C,10S,10C,9S,8H,7S,2H,2D,AS,8S	Hand #1 Straight Flush: 0 Four of a Kind: 0 Full House: 2 Straight: 0 Three of a Kind: 1 Pair: 5 Hand #2 Straight Flush: 2 6D,7D,3D,5D,4D 6D,7D,8D,5D,4D Four of a Kind: 0 Full House: 0 Straight: 2 Three of a Kind: 0 Pair: 2 Hand #3 Straight Flush: 0 Four of a Kind: 0 Full House: 0 Straight: 0 Three of a Kind: 0 Pair: 3 Average Straight Flush: 0.67 Four of a Kind: 0.00 Full House: 0.67 Straight: 0.67 Three of a Kind: 0.33 Pair: 3.33

Line Drawing

Time limit: 4 seconds

You are playing a game on a rectangular grid of unit cells with ***R*** rows and ***C*** columns ($1 \le R*C \le 5*10^5$). Each cell has a single number written on it, either 1 or 2. During a turn of the game, you are allowed to draw a single horizontal or vertical line segment on the board, starting from any location and passing through some contiguous cells (in particular, it may pass through only one cell). You gain points equal to the sum of those cells' values.

Your task is to determine, for a given value of ***P***, whether there is a way to earn exactly ***P*** points on the first turn.

Input

The first line of input contains three integers, ***R***, ***C***, and ***Q*** ($1 \le Q \le 2*10^5$), separated by single spaces. These denote, respectively, the number of rows of the grid, the number of columns, and the number of queries to answer.

The following ***R*** lines of input each contain ***C*** space-separated numbers representing the grid.

The last ***Q*** lines of input each contain a value of ***P*** ($1 \le P \le 10^6$) to consider.

Output

Your program should display exactly ***Q*** lines containing the results for successive values of ***P***. If for a given value of ***P*** it is possible to find a line that yields exactly ***P*** points on the first turn, then print the word "YES" (without quotes), and otherwise print "NO".

Sample Input 0	Sample Output 0
4 4 3 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 4 20	YES YES NO

Sample Input 1	Sample Output 1
4 4 5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 10 4 8 1	YES NO YES YES NO

Substitution Sort

Time limit: 4 seconds

Software engineering is in high demand these days, and Spongebob has decided to take an online algorithms class. Currently, he is learning about string algorithms. However, after being confused by a mess of names like Aho-Corasick and Burrows-Wheeler, Spongebob decides to take a break with a simple problem: sort a list of words in alphabetical order.

Unfortunately, Spongebob has misread the problem, and instead of rearranging the list of words so that they are in order, he mistakenly thinks he should rearrange the *alphabet* of 26 letters to accomplish his task.

In other words, Spongebob picks a simple substitution out of the 26! different permutations of letters in the alphabet, then applies this substitution to each word. For example, suppose that he chooses to substitute A → B, B → D, D → A, C → E, E → C, and all other letters to themselves. Then, the words "CAB" and "CAKE" would become "EBD" and "EBKC".

Spongebob is stumped by this new problem and needs your help. Given a list of N words of total length at most 10^6, either find a substitution that results in an alphabetically sorted list, or determine that none exists.

Note: Two strings are said to be in *alphabetical order* if the first character in which they differ is smaller in the first string than in the second, or the first string is a prefix of the second.

Input

The first line contains N, the number of words ($1 \le N \le 10^5$). The next N lines each contain a single word of the list. Words consist only of uppercase English letters (A-Z), with no spaces or special characters. Duplicate words may be present.

Output

If there exists a substitution that sorts the list, output N lines, with each new word after letter substitution on a separate line. Otherwise, print "No Solution" (without quotes).

If there are multiple solutions, your program must output the one such that if all the words in the output were concatenated in order, the resulting string would be smallest alphabetically.

Sample Input 0	Sample Output 0
4 PINEAPPLE BANANA ARTICHOKE TOMATO	ABCDFAAGD EFCFCF FHIBJKLMD ILNFIL

Sample Input 1	Sample Output 1
5 ABC ABCB ABB ABD ABD	ACB ACBC ACC ACD ACD

Sample Input 2	Sample Output 2
3 AB AA AB	No Solution

Balanced Tokens

Time limit: 4 seconds

“You’ve done it! There are no more trials!” ...is what SpongieBoi would like to hear, except our little green friend has prepared one last, extremely tough puzzle for him to overcome.

The game consists of a collection of ***N*** *tokens* ($2 \le N \le 10^5$) numbered from 1 to ***N***. Each token has a single character inscribed on it: an open parenthesis “(“, or a close parenthesis “)”. There are also *connectors*, or pieces of twine that each link two distinct tokens together. There will never be more than one connector between the same pair of tokens. Altogether, these form an undirected simple graph, with tokens representing vertices and connectors representing edges.

Originally, the puzzle starts out with no connectors. At each of the ***Q*** steps ($1 \le Q \le 3*10^5$), one of the four commands below may be executed:

1) **`connect A B`** – Add a connector between two distinct tokens numbered ***A*** and ***B***, and print out “yes”. However, Plankton is deathly afraid of cycles, so if a path of connectors linking ***A*** and ***B*** already exists when this command is executed, do not add a new connector, and print out “no” instead.
2) **`disconnect A B`** – If there exists a connector directly linking ***A*** and ***B***, remove that connector and print “yes”. Otherwise, print “no”.
3) **`toggle A`** – Change the character on token ***A***, from “(“ to ‘)”, or from “)” to “(“.
4) **`balance A B`** – Consider the path starting from token ***A*** and ending at token ***B***. If such a path does not exist, print “invalid”. Otherwise, concatenate all of the characters on tokens along the path into a string of parentheses. If this string is balanced*, then print “balanced”. Otherwise, print the minimum number of parentheses that could be added to the string (at any locations) to make it balanced.

Your task is to write a program that can evaluate these commands.

***Note:** A string of parentheses is called *balanced* if there are an equal number of “(“ and “)” characters, and all prefixes contain at least as many “(“ as “)”. For example, “()((()))” is balanced, but “)(()()” and “((())” are not.

Input

The first line of input contains two integers, ***N*** and ***Q***. The next line contains a string of ***N*** parentheses, where the ***k***-th character in the string represents what is written on the ***k***-th token. Finally, each of the next ***Q*** lines contains one of the three commands above.

Output

Output the responses to each “connect” and “balance” command in order.

Sample Input 0	Sample Output 0
3 10 ()) connect 1 2 balance 1 2 toggle 2 balance 1 2 connect 2 3 balance 2 3 connect 1 3 balance 1 1 disconnect 1 2 balance 1 3	yes balanced 2 yes balanced no 1 yes invalid

Sample Input 1	Sample Output 1
5 12))(((disconnect 3 4 connect 4 2 connect 5 4 connect 2 5 disconnect 5 4 connect 2 5 balance 5 2 balance 2 5 connect 1 4 connect 3 4 balance 5 1 balance 5 3	no yes yes no yes yes balanced 2 yes yes balanced 2