Problem A. Arena of Greed

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

Lately, Mr. Chanek frequently plays the game **Arena of Greed**. As the name implies, the game's goal is to find the greediest of them all, who will then be crowned king of Compfestnesia.

The game is played by two people taking turns, where Mr. Chanek takes the first turn. Initially, there is a treasure chest containing N gold coins. The game ends if there are no more gold coins in the chest. In each turn, the players can make one of the following moves:

- Take one gold coin from the chest.
- Take half of the gold coins on the chest. This move is only available if the number of coins in the chest is even.

Both players will try to maximize the number of coins they have. Mr. Chanek asks your help to find the maximum number of coins he can get at the end of the game if both he and the opponent plays optimally.

Input

The first line contains a single integer T $(1 \le T \le 10^5)$ denotes the number of test cases.

The next T lines each contain a single integer N $(1 \le N \le 10^{18})$.

Output

T lines, each line is the answer requested by Mr. Chanek.

Example

standard input	standard output
2 5 6	2 4

Note

For the first case, the game is as follows:

1. Mr. Chanek takes one coin.
2. The opponent takes two coins.
3. Mr. Chanek takes one coin.
4. The opponent takes one coin.

For the second case, the game is as follows:

1. Mr. Chanek takes three coins.
2. The opponent takes one coin.
3. Mr. Chanek takes one coin.
4. The opponent takes one coin.

Problem B. Blue and Red of Our Faculty!

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

It's our faculty's 34th anniversary! To celebrate this great event, the Faculty of Computer Science, University of Indonesia (Fasilkom), held CPC - Coloring Pavements Competition. The gist of CPC is two players color the predetermined routes of Fasilkom in Blue and Red. There are N Checkpoints and M undirected predetermined routes. Routes i connects checkpoint U_i and V_i, for $(1 \leq i \leq M)$. It is guaranteed that any pair of checkpoints are connected by using one or more routes.

The rules of CPC is as follows:

- Two players play in each round. One player plays as blue, the other plays as red. For simplicity, let's call these players *Blue* and *Red*.
- *Blue* will color every route in he walks on blue, *Red* will color the route he walks on red. Both players start at checkpoint number 1. Initially, all routes are gray.
- Each phase, from their current checkpoint, *Blue* and *Red* select a **different** gray route and moves to the checkpoint on the other end of the route simultaneously.
- The game ends when *Blue* or *Red* can no longer move. That is, there is no two distinct gray routes they can choose to continue moving.

Chaneka is interested in participating. However, she does not want to waste much energy. So, She is only interested in the number of final configurations of the routes after each round. Turns out, counting this is also exhausting, so Chaneka asks you to figure this out!

Two final configurations are considered different if there is a route U in a different color in the two configurations.

Input

The first line contains two integers N and M. N $(2 \leq N \leq 2 \cdot 10^3)$ denotes the number of checkpoints, M $(1 \leq M \leq 2 \cdot N)$ denotes the number of routes. It is guaranteed that every checkpoint except checkpoint 1 has exactly two routes connecting it.

The next M lines each contains two integers U_i and V_i $(1 \leq U_i, V_i \leq N, U_i \neq V_i)$, which denotes the checkpoint that route i connects.

It is guaranteed that for every pair of checkpoints, there exists a path connecting them directly or indirectly using the routes.

Output

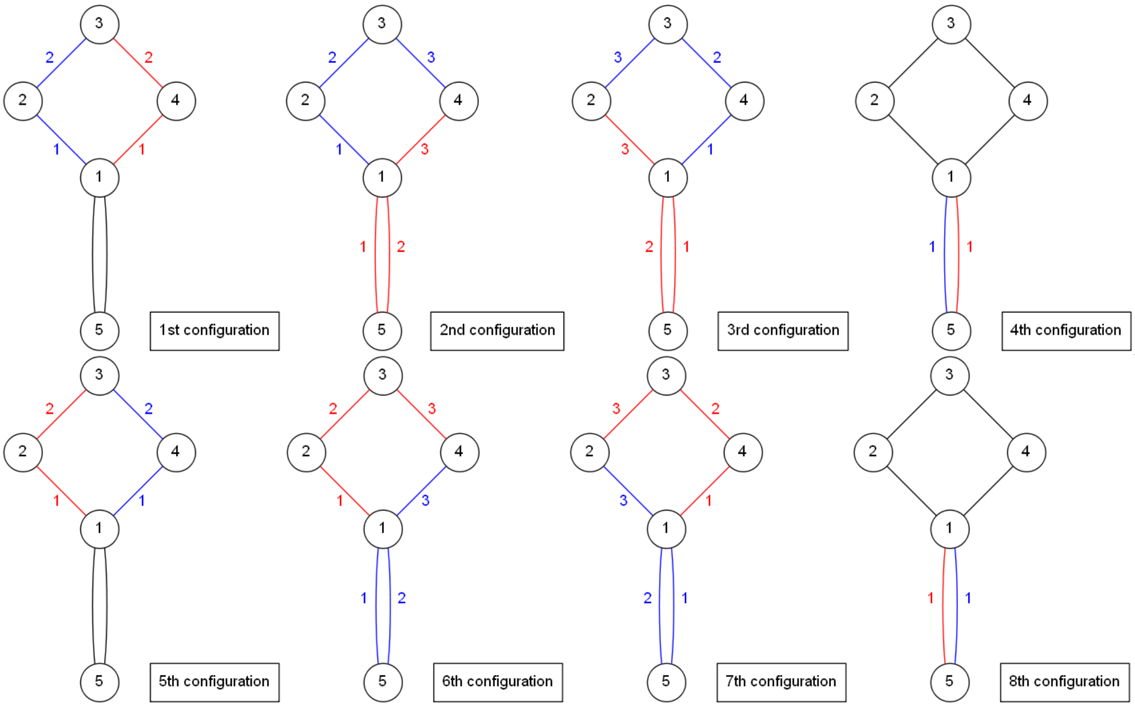
Output a single integer which denotes the number of final configurations after each round of CPC modulo $10^9 + 7$

Example

standard input	standard output
5 6 1 2 2 3 3 4 4 1 1 5 5 1	8

Note

Every possible final configuration for the example is listed below:



The blue-colored numbers give the series of moves *Blue* took, and the red-colored numbers give the series of moves *Red* took.

Problem C. Captain of Knights

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

Mr. Chanek just won the national chess tournament and got a huge chessboard of size $N \times M$. Bored with playing conventional chess, Mr. Chanek now defines a function $F(X, Y)$, which denotes the minimum number of moves to move a knight from square $(1, 1)$ to square (X, Y). It turns out finding $F(X, Y)$ is too simple, so Mr. Chanek defines:

$G(X, Y) = \sum_{i=X}^{N} \sum_{j=Y}^{M} F(i, j)$

Given X and Y, you are tasked to find $G(X, Y)$.

A knight can move from square (a, b) to square (a', b') if and only if $|a - a'| > 0$, $|b - b'| > 0$, and $|a - a'| + |b - b'| = 3$. Of course, the knight cannot leave the chessboard.

Input

The first line contains an integer T $(1 \le T \le 100)$, the number of test cases.

Each test case contains a line with four integers X Y N M $(3 \le X \le N \le 10^9, 3 \le Y \le M \le 10^9)$.

Output

For each test case, print a line with the value of $G(X, Y)$

Example

standard input	standard output
2 3 4 5 6 5 5 8 8	27 70

Problem D. Danger of Mad Snakes

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

Mr. Chanek The Ninja is one day tasked with a mission to handle mad snakes that are attacking a site. Now, Mr. Chanek already arrived at the hills where the destination is right below these hills. The mission area can be divided into a grid of size 1000×1000 squares. There are N mad snakes on the site, the i'th mad snake is located on square (X_i, Y_i) and has a danger level B_i.

Mr. Chanek is going to use the Shadow Clone Jutsu and Rasengan that he learned from Lord Seventh to complete this mission. His attack strategy is as follows:

1. Mr. Chanek is going to make M clones.
2. Each clone will choose a mad snake as the attack target. Each clone must pick a different mad snake to attack.
3. All clones jump off the hills and attack their respective chosen target at once with Rasengan of radius R. If the mad snake at square (X, Y) is attacked with a direct Rasengan, it and all mad snakes at squares (X', Y') where $max(|X' - X|, |Y' - Y|) \le R$ will die.
4. The real Mr. Chanek will calculate the score of this attack. The score is defined as the square of the sum of the danger levels of all the killed snakes.

Now Mr. Chanek is curious, what is the sum of scores for every possible attack strategy? Because this number can be huge, Mr. Chanek only needs the output modulo $10^9 + 7$.

Input

The first line contains three integers N M R $(1 \le M \le N \le 2 \cdot 10^3, 0 \le R < 10^3)$, the number of mad snakes, the number of clones, and the radius of the Rasengan.

The next N lines each contains three integers, X_i, Y_i, dan B_i $(1 \le X_i, Y_i \le 10^3, 1 \le B_i \le 10^6)$. It is guaranteed that no two mad snakes occupy the same square.

Output

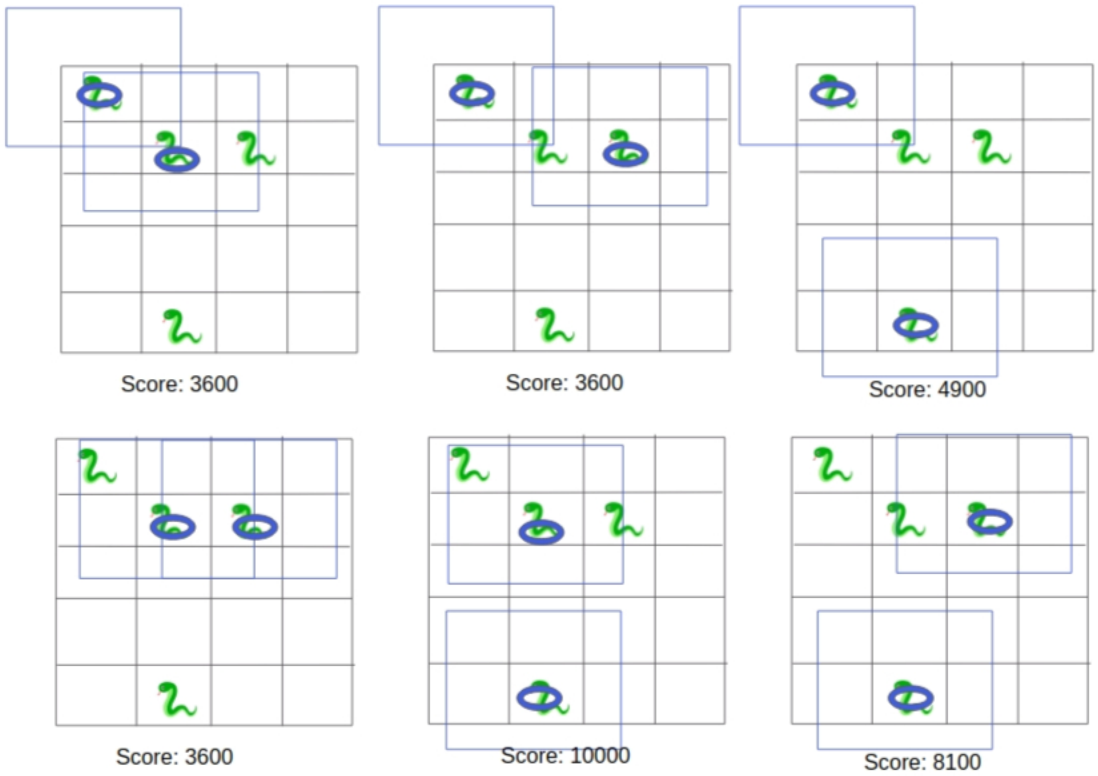
A line with an integer that denotes the sum of scores for every possible attack strategy.

Example

standard input	standard output
4 2 1 1 1 10 2 2 20 2 3 30 4 2 40	33800

Note

Here is the illustration of all six possible attack strategies. The circles denote the chosen mad snakes, and the blue squares denote the region of the Rasengan:



So, the total score of all attacks is: $3.600 + 3.600 + 4.900 + 3.600 + 10.000 + 8.100 = 33.800$.

Problem E. Excitation of Atoms

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

Mr. Chanek is currently participating in a science fair that is popular in town. He finds an exciting puzzle in the fair and wants to solve it.

There are N atoms numbered from 1 to N. These atoms are especially quirky. Initially, each atom is in normal state. Each atom can be in an excited. Exciting atom i requires D_i energy. When atom i is excited, it will give A_i energy. You can excite any number of atoms (including zero).

These atoms also form a peculiar one-way bond. For each i, $(1 \le i < N)$, if atom i is excited, atom E_i will also be excited at no cost. Initially, $E_i = i + 1$. Note that atom N cannot form a bond to any atom.

Mr. Chanek must change **exactly** K bonds. Exactly K times, Mr. Chanek chooses an atom i, $(1 \le i < N)$ and changes E_i to a different value other than i and the current E_i. Note that an atom's bond can remain unchanged or changed more than once. Help Mr. Chanek determine the maximum energy that he can achieve!

note: You must first change **exactly** K bonds before you can start exciting atoms.

Input

The first line contains two integers N K $(4 \le N \le 10^5, 0 \le K < N)$, the number of atoms, and the number of bonds that must be changed.

The second line contains N integers A_i $(1 \le A_i \le 10^6)$, which denotes the energy given by atom i when on excited state.

The third line contains N integers D_i $(1 \le D_i \le 10^6)$, which denotes the energy needed to excite atom i.

Output

A line with an integer that denotes the maximum number of energy that Mr. Chanek can get.

Example

standard input	standard output

`6 1`
`5 6 7 8 10 2`
`3 5 6 7 1 10`
	`35`

Note

An optimal solution to change E_5 to 1 and then excite atom 5 with energy 1. It will cause atoms 1, 2, 3, 4, 5 be excited. The total energy gained by Mr. Chanek is (5 + 6 + 7 + 8 + 10) - 1 = 35.

Another possible way is to change E_3 to 1 and then exciting atom 3 (which will excite atom 1, 2, 3) and exciting atom 4 (which will excite atom 4, 5, 6). The total energy gained by Mr. Chanek is (5 + 6 + 7 + 8 + 10 + 2) - (6 + 7) = 25 which is not optimal.

Problem F. Flamingoes of Mystery

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

This is an interactive problem. You have to use a ***`flush`*** *operation right after printing each line. For example, in C++ you should use the function* ***`fflush(stdout)`****, in Java —* ***`System.out.flush()`****, in Pascal —* ***`flush(output)`*** *and in Python —* ***`sys.stdout.flush()`****.*

Mr. Chanek wants to buy a flamingo to accompany his chickens on his farm. Before going to the pet shop, Mr. Chanek stops at an animal festival to have fun. It turns out there is a carnival game with a flamingo as the prize.

There are N mysterious cages, which are numbered from 1 to N. Cage i has A_i ($0 \le A_i \le 10^3$) flamingoes inside ($1 \le i \le N$). However, the game master keeps the number of flamingoes inside a secret. To win the flamingo, Mr. Chanek must guess the number of flamingoes in each cage.

Coincidentally, Mr. Chanek has N coins. Each coin can be used to ask once, what is the total number of flamingoes inside cages numbered L to R inclusive? With $L < R$.

Input

Use standard input to read the responses of your questions.

Initially, the judge will give an integer N ($3 \le N \le 10^3$), the number of cages, and the number of coins Mr. Chanek has.

For each of your questions, the jury will give an integer that denotes the number of flamingoes from cage L to R inclusive.

If your program does not guess the flamingoes or ask other questions, you will get "Wrong Answer". Of course, if your program asks more questions than the allowed number, your program will get "Wrong Answer".

Output

To ask questions, your program must use standard output.

Then, you can ask at most N questions. Questions are asked in the format "? L R", ($1 \le L < R \le N$).

To guess the flamingoes, print a line that starts with "!" followed by N integers where the i-th integer denotes the number of flamingo in cage i. After answering, your program must terminate or will receive the "idle limit exceeded" verdict. You can only guess the flamingoes once.

Examples

standard input	standard output
6	
	? 1 2
5	
	? 5 6
15	
	? 3 4
10	
	! 1 4 4 6 7 8

Note

In the sample input, the correct flamingoes amount is $[1, 4, 4, 6, 7, 8]$.

Problem G. Greasy Thief of The Village

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

Mr. Chanek is police in a village. His town can be represented as a grid G of size $N \times M$, where G_{ij} can be either "." or "#", which denotes a passable area and unpassable area (walls, houses, and others) respectively. The outside of the grid is assumed to be unpassable areas.

One day, a thief makes a commotion in the village, and Mr. Chanek is trying to apprehend him. Each second, both the thief and Mr. Chanek can move one square to eight possible directions (up, down, left, right, left-up, left-down, right-up, right-down) with the movement is assumed to happen instantly. As an example, assume Mr. Chanek current position is as follows:

```
.##
#C.
...
```

Then Mr. Chanek can move to 5 directions, which is: left-up, left-down, down, right-down, and right.

Here is the chase scenario:

- At second i exactly, the thief can move 1 square.
- At second $(i + 0.5)$, Mr. Chanek can move 1 square.

Both Mr. Chanek and the thief can choose not to move at a second.

If the thief and Mr. Chanek move optimally, determine whether Mr. Chanek will apprehend the thief, or the thief can forever avoid Mr. Chanek. The thief is said to be caught by Mr. Chanek if they occupy the same square.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^3$), the number of rows and columns of the grid, respectively.

The next N lines each contains M characters. The j'th character on the i'th line denotes G_{ij} ($G_{ij} \in \{"C", "M", ".", "\#"\}$). There will be exactly one character 'C' and 'M' in G, representing Mr. Chanek's and the thief's initial position respectively.

It is guaranteed that there exist a way to visit every pair of non-'#' square in G.

Output

Output "tertangkap" (without quotes, Indonesian for "captured") if Mr. Chanek can apprehend the thief. Output "lolos" (without quotes, Indonesian for "evaded") if the thief can avoid Mr. Chanek indefinitely.

Examples

standard input	standard output
5 5 ..#.# .##C. M.... ....#..	tertangkap
4 7 ##.M.## ##.#.## ##.C.## #######	lolos

Note

here are the possible movements of the first example:

- If the thief moves up:
  1. Mr. Chanek moves left-down.
  2. The thief must move down to avoid capture.
  3. Mr. Chanek moves left.
  4. Whether the thief moves up or down, Mr. Chanek will apprehend him in the next second.
- If the thief moves down:
  1. Mr. Chanek moves left-down.
  2. Mr. Chanek will catch the thief if he moves up, left-up, or right (the first case). So the thief can only move down or right-down.
  3. Whether the thief moves right or right-down, Mr. Chanek will move left-down and apprehend the thief the next second.

So, we can see that the thief cannot avoid Mr. Chanek forever.

For the second example, the thief can avoid Mr. Chanek by moving the square that maximizes the distance between him and Mr. Chanek.

Problem H. Huge Boxes of Animal Toys

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

Chaneka has a hobby of playing with animal toys. Every toy has a different fun value, a real number. Chaneka has four boxes to store the toys with specification:

- The first box stores toys with fun values in range of $(\infty, -1]$.
- The second box stores toys with fun values in range of $(-1, 0)$.
- The third box stores toys with fun values in range of $(0, 1)$.
- The fourth box stores toys with fun value in range of $[1, \infty)$.

Chaneka has A, B, C, D toys in the first, second, third, and fourth box, respectively. One day she decides that she only wants one toy, a super toy. So she begins to create this super toy by sewing all the toys she has.

While the number of toys Chaneka has is more than 1, she takes two different toys randomly and then sews them together, creating a new toy. The fun value of this new toy is equal to the multiplication of fun values of the sewn toys. She then puts this new toy in the appropriate box. She repeats this process until she only has one toy. This last toy is the super toy, and the box that stores this toy is the special box.

As an observer, you only know the number of toys in each box initially but do not know their fun values. You also don't see the sequence of Chaneka's sewing. Determine which boxes can be the special box after Chaneka found her super toy.

Input

The first line has an integer T $(1 \le T \le 5 \cdot 10^4)$, the number of test cases.

Every case contains a line with four space-separated integers A B C D $(0 \le A, B, C, D \le 10^6, A + B + C + D > 0)$, which denotes the number of toys in the first, second, third, and fourth box, respectively.

Output

For each case, print four space-separated strings. Each string represents the possibility that the first, second, third, and fourth box can be the special box from left to right.

For each box, print "Ya" (Without quotes, Indonesian for yes) if that box can be the special box. Print "Tidak" (Without quotes, Indonesian for No) otherwise.

Example

standard input	standard output

`2`
`1 2 0 1`
`0 1 0 0`
	`Ya Ya Tidak Tidak`
	`Tidak Ya Tidak Tidak`

Note

For the first case, here is a scenario where the first box is the special box:

- The first box had toys with fun values $\{-3\}$.

- The second box had toys with fun values $\{-0.5, -0.5\}$
- The fourth box had toys with fun values $\{3\}$

The sewing sequence:

1. Chaneka sews the toy with fun -0.5 and -0.5 to a toy with fun 0.25 and then put it in the third box.
2. Chaneka sews the toy with fun -3 and 0.25 to a toy with fun -0.75 and then put it in the second box.
3. Chaneka sews the toy with fun -0.75 and 3 to a toy with fun -1.25 and then put it in the first box, which then became the special box.

Here is a scenario where the second box ends up being the special box:

- The first box had toys with fun values $\{-3\}$
- The second box had toys with fun values $\{-0.33, -0.25\}$.
- The fourth box had toys with fun values $\{3\}$.

The sewing sequence:

1. Chaneka sews the toy with fun -3 and -0.33 to a toy with fun 0.99 and then put it in the third box.
2. Chaneka sews the toy with fun 0.99 and 3 to a toy with fun 2.97 and then put in it the fourth box.
3. Chaneka sews the toy with fun 2.97 and -0.25 to a toy with fun -0.7425 and then put it in the second box, which then became the special box.

There is only one toy for the second case, so Chaneka does not have to sew anything because that toy, by definition, is the super toy.

Problem I. Impressive Harvesting of The Orchard

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	256 megabytes

Mr. Chanek has an orchard structured as a rooted ternary tree with N vertices numbered from 1 to N. The root of the tree is vertex 1. P_i denotes the parent of vertex i, for $(2 \leq i \leq N)$. Interestingly, the height of the tree is not greater than 10. Height of a tree is defined to be the largest distance from the root to a vertex in the tree.

There exist a bush on each vertex of the tree. Initially, all bushes have fruits. Fruits will not grow on bushes that currently already have fruits. The bush at vertex i will grow fruits after A_i days since its last harvest.

Mr. Chanek will visit his orchard for Q days. In day i, he will harvest all bushes that have fruits on the subtree of vertex X_i. For each day, determine the sum of distances from every harvested bush to X_i, and the number of harvested bush that day. Harvesting a bush means collecting **all** fruits on the bush.

For example, if Mr. Chanek harvests all fruits on subtree of vertex X, and harvested bushes $[Y_1, Y_2, \ldots, Y_M]$, the sum of distances is $\sum_{i=1}^{M} \text{distance}(X, Y_i)$

$\text{distance}(U, V)$ in a tree is defined to be the number of edges on the simple path from U to V.

Input

The first line contains two integers N and Q $(1 \leq N,\ Q, \leq 5 \cdot 10^4)$, which denotes the number of vertices and the number of days Mr. Chanek visits the orchard.

The second line contains N integers A_i $(1 \leq A_i \leq 5 \cdot 10^4)$, which denotes the fruits growth speed on the bush at vertex i, for $(1 \leq i \leq N)$.

The third line contains $N-1$ integers P_i $(1 \leq P_i \leq N, P_i \neq i)$, which denotes the parent of vertex i in the tree, for $(2 \leq i \leq N)$. It is guaranteed that each vertex can be the parent of at most 3 other vertices. It is also guaranteed that the height of the tree is not greater than 10.

The next Q lines contain a single integer X_i $(1 \leq X_i \leq N)$, which denotes the start of Mr. Chanek's visit on day i, for $(1 \leq i \leq Q)$.

Output

Output Q lines, line i gives the sum of distances from the harvested bushes to X_i, and the number of harvested bushes.

Examples

standard input	standard output
2 3 1 2 1 2 1 1	0 1 0 1 1 2
5 3 2 1 1 3 2 1 2 2 1 1 1 1	6 5 3 2 4 4

Note

For the first example:

- On day 1, Mr. Chanek starts at vertex 2 and can harvest the bush at vertex 2.
- On day 2, Mr. Chanek starts at vertex 1 and only harvest from bush 1 (bush 2's fruit still has not grown yet).
- On day 3, Mr. Chanek starts at vertex 1 and harvests the fruits on bush 1 and 2. The sum of distances from every harvested bush to 1 is 1.

For the second example, Mr. Chanek always starts at vertex 1. The bushes which Mr. Chanek harvests on day one, two, and three are $[1, 2, 3, 4, 5], [2, 3], [1, 2, 3, 5]$, respectively.