Problem A. Linked List and Xors

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

NOTE - you can create a singly/doubly linked list from scratch. Your submission will not be graded if implementation of singly/doubly linked list is not found.

You will be given a linked list with N elements. Following are some operations that can be performed on the linked list:

- **KthNode K :** Print -1 if K is greater than the length of the linked list. Otherwise in this operation you have to print the K^{th} node of the linked list.
- **Delete K :** Print -1 if K is greater than the length of the linked list. Otherwise in this operation you have to delete the K^{th} node of the linked list.
- **Add K X:** Print -1 if K is greater than the length of the linked list. Otherwise in this operation you have to Add the value X to the K^{th} node of the linked list.
- **Subtract K X:** Print -1 if K is greater than the length of the linked list. Otherwise in this operation you have to Subtract the value X to the K^{th} node of the linked list.
- **Xor K :** Print -1 if K is greater than the length of the linked list. Otherwise in this operation you have to print the value of the xor of last K^{th} nodes of the linked list.

You will be given Q of the above mentioned operations to be performed on the linked list.

Constraints:

- $1 \leq N * Q \leq 10^7$
- $1 \leq K \leq 10^5$
- $1 \leq$ Linked List Node value, X $\leq 10^9$

Input

The first line contains 2 integers N and Q denoting the length of the linked list and number of queries.
The next line contains N spaced integers denoting the elements of the linked list.
Each of the following Q lines contain one of the operations to be performed on the linked list as mentioned in the problem.

Output

Print the output corresponding to the operations if any.

Examples

standard input	standard output
10 9 3 10 3 3 5 7 1 2 8 6 Delete 9 KthNode 1 Delete 4 KthNode 6 KthNode 5 Delete 5 KthNode 2 Xor 7 Xor 5	3 1 7 10 10 3
4 4 4 5 10 5 Delete 2 Subtract 3 2 Xor 4 Delete 1	-1

Problem B. Binary String Search Trees

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

NOTE - you need to implement a BST from scratch. Also note that you are not allowed to use libraries other than stdlib.h and stdio.h your submission will not be graded if implementation of BST is not found.

Binary Search Trees are binary trees following a special property where in the key of each internal node being greater than all the keys in the respective node's left subtree and less than the ones in its right subtree.

Following the same definition we have a come up with a new BST called **Binary String Search Trees**. It's a BST where in each key would be string. The keys would be kept in the left/right subtrees by checking whether they are smaller/greater than the key of the parent node.

A string a is smaller than string b if the first non similar character in both strings are lexicographically smaller/greater than each other.

NOTE: for example if strings are "aabb" and "aa", then "aabb" would be greater than "aa". But if strings are "aabb" and "ab", then "ab" would be greater than "aabb".

You will be given N unique strings at first. You will be required to create a BST by comparing the strings while inserting them. The strings will be inserted into the BST one by one starting in the same order as given in input.

You can perform 2 operations on the BST constructed.

- **Delete pref :** Print -1 if no node with the prefix "pref" is deleted from the BST. Otherwise delete nodes with the prefix "pref" and don't print anything.
- **Find pref :** Print -1 if no node with the prefix "pref" exists in the BST. Otherwise print the strings with prefix "pref" in lexicographically sorted order.

You will be given Q of the above mentioned operations to be performed on the BST.

Constraints:

- $1 \leq N * Q \leq 10^6$
- $1 \leq$ length of string, pref ≤ 20
- Note that the strings would contain lowercase letters from $a - z$.

Input

The first line contains 2 integers N and Q denoting the number of the elements in the BST and number of queries.
The next N lines contains N unique strings.
Each of the following Q lines contain one of the operations to be performed on the BST as mentioned.

Output

Print output corresponding to the operation mentioned above.

Examples

standard input	standard output
3 1 aba abc abd Find ab	aba abc abd
3 2 aba abc abd Delete cd Find aba	-1 aba

Problem C. File System

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

Implement File System using Linked List.

You need to write the following functions that are used in the file system.

S.No.	Command	Description
1	pwd	Prints the path of the directory starting from root
2	ls	Lists all directories inside the current directory (space-separated) sorted in lexi
3	mkdir 'directory_name'	Create a directory 'directory_name'
4	cd 'directory_name'	Change current directory to 'directory_name'
5	cd ..	Change current directory to its parent directory
6	rm -r 'directory_name'	Deletes 'directoryName' and all its subsequent directories

Note:

- **Initially you are at 'root' directory.**
- **Each Directory can have maximum** 20 **directories**

You have to perform Q **operations in the File System.**

Input

The first line of the input contains a single integer Q **(**$1 <= Q <= 10^5$**)— the number of operations to perform.**

Each of the next Q **lines contains two lines:**

- **The first line contains an integer** n **— the number of words in the command.**
- **The second line contains the command as a string of** n **words as described above.**

Output

You have to print the output the each command as follows

S.No.	Command	Output
1	**pwd**	**Prints all directories starting from the root each directory separa**
2	**ls**	**dir1 dir2 dir3**
3	**mkdir 'directory_name'**	**No Output if successfull otherwise -1**
4	**cd 'directory_name'**	**No Output if successfull otherwise -1**
5	**rm -r 'directory_name'**	**No Output if successfull otherwise -1**

Examples

standard input	standard output
10 2 mkdir dir1 2 cd dir1 1 pwd 2 cd dir2 2 cd .. 1 pwd 2 mkdir dir2 1 ls 3 rm -r dir2 1 ls	root/dir1 -1 root dir1 dir2 dir1
12 2 mkdir dir1 2 mkdir dir2 2 mkdir dir3 1 ls 2 cd dir2 2 mkdir dir1 1 pwd 2 cd dir1 1 pwd 2 mkdir dir3 2 mkdir dir4 1 ls	dir1 dir2 dir3 root/dir2 root/dir2/dir1 dir3 dir4
6 2 mkdir dir1 2 mkdir dir2 3 rm -r dir2 1 ls	dir1 -1 -1

Note

Explanation of Testcase-1

Bold words in explanation represent output to be printed

Initially, you are at the root directory.

mkdir dir1	**create directory dir1 inside root directory**
cd dir1	**change current working directory from root to dir1 directory**
pwd	**print the current working directory that is root/dir1.**
cd dir2	**since dir1 doesn't have a directory named dir2, the operation is unsuccessful and henc**
cd ..	**the working directory location changes to the parent directory of dir1 which is the roo**
pwd	**print the current working directory which is root.**
mkdir dir2	**create directory dir2 inside the root directory.**
ls	**list all the directories inside the current working directory. i.e. dir1 dir2.**
rm -r dir2	**remove directory dir2.**
ls	**Since we have deleted dir2, therefore we only left with dir1 and hence it prints the ou**

Problem D. Binary Tree Height-Diameter Difference

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

You are given the level order traversal of a binary tree. The level order traversal of a binary tree is defined as the sequence of node values visited by level, from the root node at level 0, followed by the nodes at level 1, then level 2, and so on. Each level is visited from left to right. If a node has no child, it is represented by the integer -1 in the level order traversal sequence.

Your task is to compute the absolute difference between the height and diameter of the binary tree represented by the given level order traversal.

The height of a binary tree is defined as the length of the longest path from the root node to any leaf node in the tree.

The diameter of a binary tree is defined as the length of the longest path between any two nodes in the tree.

Input

The input consists of two lines. The first line contains a single integer n ($1 \leq n \leq 10^4$), the number of nodes in the binary tree.

The second line contains $2n+1$ space-separated integers representing the level order traversal of the binary tree, where each integer represents the value of a node in the tree.

The values of the nodes are unique and range from 1 to n. If a node has no child, it is represented by the integer -1.

Output

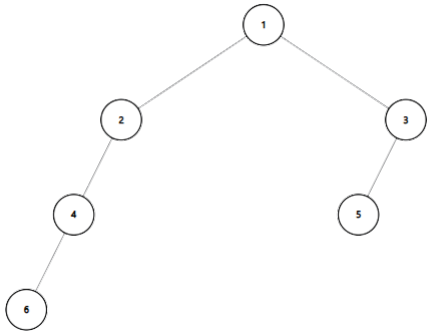
Output a single integer, the absolute difference between the height and diameter of the binary tree

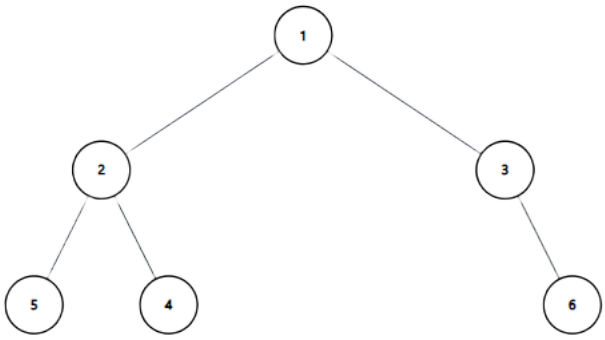
Examples

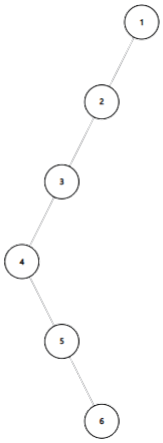
standard input	standard output
6 1 2 3 4 -1 5 -1 6 -1 -1 -1 -1 -1	2
6 1 2 3 5 4 -1 6 -1 -1 -1 -1 -1 -1	2
6 1 2 -1 3 -1 4 -1 -1 5 -1 6 -1 -1	0

Note

Binary Tree visualization for Testcase-1, Testcase-2 and Testcase-3 respectively.


1
2
3
4
5
6


1
2
3
5
4
6


1
2
3
4
5
6