1. Each time Sunny and Johnny take a trip to the Ice Cream Parlour, they pool their money to buy ice cream. On any given day, the parlour offers a line of flavours. Each flavour has a cost associated with it. Given the value of money and the cost of each flavour for trips to the Ice Cream Parlour, help Sunny and Johnny choose two distinct flavours such that they spend their entire pool of money during each visit. ID numbers of the ice creams are *1- based* index numbers associated with a cost. For each trip to the parlour, print the ID numbers for the two types of ice cream that Sunny and Johnny purchase as two space-separated integers on a new line. You must print the smaller ID first and the larger ID second.

Input:
The first 3 input lines are as follows:
The first line contains total amount money.
The second line contains an integer, **n**, the size of the array cost.
The third line contains **n** space-separated integers denoting the **cost[i]**.

Output:
Print two space-separated integers denoting the respective indices for the two distinct flavours they choose to purchase in ascending order. Note that multiple answers may be possible. In that case, print only one answer.

Sample Input:
7
6
1 5 4 6 9 7

Sample Output:
1 6

2. Harold is a kidnapper who wrote a ransom note, but now he is worried it will be traced back to him through his handwriting. He found a magazine and wants to know if he can cut out whole words from it and use them to create an untraceable replica of his ransom note. The words in his note are *case-sensitive* and he *must* use only whole words available in the magazine. He *cannot* use substrings or concatenation to create the words he needs.
Given the words in the magazine and the words in the ransom note, print Yes if he can replicate his ransom note *exactly* using whole words from the magazine; otherwise, print No.
For example, the note is "Attack at dawn". The magazine contains only "attack at dawn". The magazine has all the right words, but there's a case mismatch. The answer is No.

Input:
The first line contains two space-separated integers, **m** and **n**, the numbers of words in the magazine and the note.
The second line contains **m** space-separated strings, each is **magazine[i]**.
The third line contains **n** space-separated strings, each **note[i]**.

Output:
Print Yes if he can use the magazine to create an untraceable replica of his ransom note. Otherwise, print No.

Example:
6 4
give me one grand today night
give one grand today

Answer:
Yes

3. Given a Binary Tree, write a function that returns the size of the largest subtree which is also a Binary Search Tree in terms of the number of nodes present in it. If the complete Binary Tree is BST, then return the size of the whole tree.

Input:
First line contains number of nodes, **n**.
The next **n-1** lines contain value of nodes which are connected.
Last line contains root.

Output:
Size of the maximum subtree which is BST.
All the nodes which are in the subtree.

Example:
5
5 2
5 4
2 1
2 3
5

Answer:
3
2 1 3

4. Shantam is very rich. He is extremely talented in almost everything except in mathematics. So, one day, he pays a visit to a temple (to pray for his upcoming mathematics exams) and decides to donate some amount of money to the poor people (everyone is poor on a relative scale to Shantam). He orders **N** people to sit in a linear arrangement and indexes them from **1** to **N**, to ease the process of donating money. As with all rich people, their way of doing things is completely different. Shantam donates his money in **M** steps, each of his step being to choose two indices **L** and **R**, and an amount of money **C** and then he donates **C** currencies to each and every person whose index lies in **[L, R]**. In other words, he donates **C** currencies to every index **i** such **L <= i <= R**.

Luckily, you too were among the **N** people selected and somehow you know all the **M** steps in advance. Figure out the maximum amount of money you can obtain and at what position you should sit to get this maximum amount of money. If multiple positions guarantee the maximum amount of money, output the minimum index out of all these possibilities.

You will be given initial **L[1], R[1] and C[1]** (which points to step 1) as well as **P , Q** and **S** (integer numbers to calculate **L, R** and **C** for steps **i > 1**). Each subsequent step (**i > 1**) is generated as:

$L[i] = (L[i-1] * P + R[i-1]) \% N + 1;$
$R[i] = (R[i-1] * Q + L[i-1]) \% N + 1;$

```
if(L[i] > R[i])
        swap(L[i] , R[i]);
C[i] = (C[i-1] * S) % 1000000 + 1;
```

Input:
The first line contains T, the number of test cases. The first line of each test case contains two space separated integers N and M, which denotes the number of people and the number of steps, respectively. The next line contains integers L, R, C, P, Q and S, which are used to generate the queries using the method specified above.

Output:
For each test case, output one line containing two space separated integers, the first being the optimal position and the second being the highest amount of money that can be obtained.

Example:
2
5 2
1 1 1 1 1 1
5 1
1 3 2 4 3 5

Answer:
3 2
1 2

5. Given a Binary Search Tree (BST) and a range *[min, max]*, remove all keys which are outside the given range. The modified tree should also be BST.

Input-
First line contains number of nodes, *n*.
The next *n-1* lines contain value of nodes which are connected.
The next line contains root.
The last line contains the range *[min, max]*.

Output-
Output the in-order traversal of the final binary search tree.

Example:
7
6 -13
6 14
-13 -8
14 13
14 15
13 7
6
-10 13

Answer:
6 -8 13 7

6. You are the owner of a library. Now you need to setup a search engine for all the books in the library. Your search engine should also be able to auto correct few mistakes done by user.

Given a list of book names, we want to implement a spellchecker that converts a query word into a correct word. For a given query word, the spell checker handles two categories of spelling mistakes:

Capitalization: If the query matches a word in the book names (case-insensitive), then the query word is returned with the same case as the case in the wordlist.
- Example: books= ["yellow"], query = "YellOw": correct = "yellow"
- Example: books= ["Yellow"], query = "yellow": correct = "Yellow"
- Example: books= ["yellow"], query = "yellow": correct = "yellow"

Vowel Errors: If after replacing the vowels ('a', 'e', 'i', 'o', 'u') of the query word with any vowel individually, it matches a word in the wordlist (case-insensitive), then the query word is returned with the same case as the match in the wordlist.
- Example: books= ["YellOw"], query = "yollow": correct = "YellOw"
- Example: books= ["YellOw"], query = "yeellow": correct = "" (no match)
- Example: books= ["YellOw"], query = "yllw": correct = "" (no match)

In addition, the spell checker operates under the following precedence rules:
- When the query exactly matches a word in the wordlist (case-sensitive), you should return the same word back.
- When the query matches a word up to some capitalization errors, you should return the first such match in the wordlist.
- When the query matches a word up to some vowel errors, you should return the first such match in the wordlist.
- If the query has no matches in the wordlist, you should return the empty string.

Given some queries, return a list of words answer, where **answer[i]** is the correct word for **query = queries[i].**

Sample Input Format:
N - number of books
N lines contain the name of books
Q - number of queries
Q lines for each query word

Sample Input:
4
KiTe
kite
hare
Hare
10
kite
Kite
KiTe
Hare
HARE
Hear
hear
keti
keet
keto

Sample Output:
kite
KiTe
KiTe
Hare
hare
……
……
KiTe
……
KiTe

7. Mess 1 is working on its menu list. After recent cases it has decided to create a dataset in which they are asking the patients of the most recent item they ate. Based on that they want to find the **k** most frequent dish resulting in food poisoning.

Note - Please answer in sorted order of frequency

Sample Input
6 (Number of patients)
Dosa
Dosa
Bread Butter
Pav bhaji
Pav bhaji
Poha
2

Sample Output
Dosa
Pav bhaji

8. Hogwarts is under attack. Harry is making a Dumbledore army to fight back. He has arranged **N** people and he knows their current power levels. Now in this fight, combinations are very important. Given an array of power levels, find out whether there are two distinct indices **i** and **j** (**1 <= i, j < N**) in the array such that the absolute difference between **power[i]** and **power[j]** is at most **t** and the absolute difference between **i** and **j** is at most **k**. If the above conditions are satisfied, print "Hogwarts wins" or else print "Voldemort wins".
Note - Power level can also be a negative number.

Sample Input Format:
N k t
An array of **N** integers

Sample Input:
4 3 0
1 2 3 1

Sample Output:
Hogwarts Wins

Sample Input:

6 2 3
1 5 9 1 5 9

Sample Output:
Voldemort Wins

9. Given pre-order and in-order traversals of a binary search tree, construct the binary search tree. Output the post-order of the tree.

Note - You need to construct the tree and then print the post-order
Note - You can assume that there are no duplicates

Sample Input Format:
N
In-order traversal
Pre-order traversal

Sample Input:
3
1 2 3
2 1 3

Sample Output:
2 3 1

10. You are the word master. The words are just too big for you. You decide to make a shortest possible unique prefix for every word.

Sample Input:
4 (Number of words)
Zebra
Dog
Duck
Dove

Sample Output:
Z
Dog
Du
Dov

****