# Tree Traversal

Q. Your friend wants to play game with you, he ask you to find his current location by the given below information.

The tree was contructed by splitting level wise first left and then right the person can move one step at a time . By the given input where his location at nth step.

For example

input

1 2 3 4 5

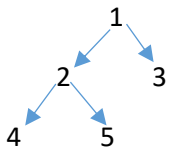
1

3

Output

5

The tree constructed looks like




Input description

First line contains list of places and second line has integer value (1=inorder, 2=preorder, 3=postorder) and last line contains number of steps n.

Output description

Print the result

## Test cases

1. Input
   Chennai Mumbai Delhi Kolkata Gujarat
   1
   3
   Output
   Gujarat

2. Input
   India Pakistan Australia America Africa Brazil
   2

4
Output
Africa

3. Input
North West East South
3
4
Output
North

4. Input
Vadapalani Porur Koyambedu Kundrathur Adayar Vadapalani Porur Vadapalani
1
1
Output
Vadapalan

5. Input
Dubai Malaysia Kodaikanal
3
3
Output
Dubai

6. Input
Delhi
2
4
Output
Not found
7. Input
Dubai Malaysia Kodaikanal
5
2
Output
Not found

## Source code

```
class newNode:

    def __init__(self, data):
```

```
        self.data = data
        self.left = self.right = None


def insertLevelOrder(arr, root, i, n):


    if i < n:
        temp = newNode(arr[i])
        root = temp


        root.left = insertLevelOrder(arr, root.left, 2 * i + 1, n)


        root.right = insertLevelOrder(arr, root.right, 2 * i + 2, n)
    return root


def inOrder(root):
    if root != None:
        inOrder(root.left)
```

```
            ans.append(root.data)
            inOrder(root.right)
def preOrder(root):
    if root != None:
            ans.append(root.data)
            preOrder(root.left)
            preOrder(root.right)
def postOrder(root):
    if root != None:
            postOrder(root.left)
            postOrder(root.right)
            ans.append(root.data)

arr = list(input().split())
type_of_traversal=int(input())
jump=int(input())
n = len(arr)
root = None
root = insertLevelOrder(arr, root, 0, n)
ans=[]
if(type_of_traversal==1):
   inOrder(root)
```

```
elif(type_of_traversal==2):
    preOrder(root)
elif(type_of_traversal==3):
    postOrder(root)

if(jump>n or type_of_traversal>3 or type_of_traversal<1):
    print('Not found')
else:
    print(ans[jump-1])
```

GOVARDHANAN P