**Rearrange an array with O(1) extra space :-**

Medium Accuracy: 56.34% Submissions: 96K+ Points: 4

Given an array **arr[]** of size **N** where every element is in the range from **0 to n-1**. Rearrange the given array so that the transformed array **arrT[i]** becomes **arr[arr[i]]**.

**NOTE:** **arr**and **arrT**are both same variables, representing the array before and after transformation respectively.

**Example 1:**

**Input:**

N = 2

arr[] = {1,0}

**Output:** 0 1

**Explanation:**

arr[arr[0]] = arr[1] = 0.

arr[arr[1]] = arr[0] = 1.

**Example 2:**

**Input:**

N = 5

arr[] = {4,0,2,1,3}

**Output:** 3 4 2 0 1

**Explanation:**

arr[arr[0]] = arr[4] = 3.

arr[arr[1]] = arr[0] = 4.

and so on.

**Your Task:**  
You don't need to read input or print anything. The task is to complete the function **arrange**() which takes arr and N as input parameters and rearranges the elements in the array in-place.

**Expected Time Complexity:**O(N)  
**Expected Auxiliary Space:**O(1)

**Constraints:**  
1 <= N <= 105  
0 <= Arr[i] < N

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

// arr: input array

// n: size of array

//Function to rearrange an array so that arr[i] becomes arr[arr[i]]

//with O(1) extra space.

void arrange(long long arr[], int n) {

//new item stored = (array size \* new actual refered item) + old item stored

//new item actual = new item stored / array size

//old item stored = new item stored % array size

for(int i=0; i<n; i++){

arr[i] = (n \* (arr[arr[i]] % n)) + arr[i];

}

for(int i=0; i<n; i++){

arr[i] = arr[i] / n;

}

return;

}

};

//{ Driver Code Starts.

int main(){

int t;

//testcases

cin>>t;

while(t--){

int n;

//size of array

cin>>n;

long long A[n];

//adding elements to the array

for(int i=0;i<n;i++){

cin>>A[i];

}

Solution ob;

//calling arrange() function

ob.arrange(A, n);

//printing the elements

for(int i=0;i<n;i++){

cout << A[i]<<" ";

}

cout<<endl;

}

return 0;

}

// } Driver Code Ends

**T.C :- O(n)**

**S.C :- O(1)**