COMPETITIVE PROGRAMMING

1. Finding Duplicates-O(n^2) Time Complexity, O(1) Space Complexity

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

```c
#include<stdio.h>
int main(){
    int a;scanf("%d",&a);int arr[a];
    for(int i=0;i<a;i++)scanf("%d",&arr[i]);
    for(int i=0;i<a-1;i++){
        for(int j=i;j<a;j++){
            if(arr[i]>arr[j]){
                int temp=arr[i];arr[i]=arr[j];arr[j]=temp;}}}
    for(int i=1;i<a;i++){
        if (arr[i]==arr[i-1]){
            printf("%d",arr[i]);break;}}}
```

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

2. Finding Duplicates-O(n) Time Complexity, O(1) Space Complexity

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

```
#include<stdio.h>
int main(){
    int arr[100],a,b;scanf("%d",&a);
    for(int i=0;i<a;i++){
        scanf("%d",&b);
        if(!arr[b])arr[b]=1;
        else printf("%d",b);}}
```

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔