import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.regex.\*;

public class Solution {

// Complete the sockMerchant function below.

static int sockMerchant(int n, int[] ar) {

HashMap<Integer,Integer> hm=new HashMap<>();

int count = 0;

for(int i=0; i<ar.length;i++){

if(!hm.containsKey(ar[i])){

hm.put(ar[i],1);

}else{

hm.put(ar[i],hm.get(ar[i])+1);

}

}

for(int i=0; i<ar.length;i++){

if(hm.containsKey(ar[i]) && hm.get(ar[i])>=2){

hm.put(ar[i],hm.get(ar[i])-2);

count++;

continue;

}

}

return count;

}

private static final Scanner scanner = new Scanner(System.in);

public static void main(String[] args) throws IOException {

BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

int n = scanner.nextInt();

scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

int[] ar = new int[n];

String[] arItems = scanner.nextLine().split(" ");

scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

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

int arItem = Integer.parseInt(arItems[i]);

ar[i] = arItem;

}

int result = sockMerchant(n, ar);

bufferedWriter.write(String.valueOf(result));

bufferedWriter.newLine();

bufferedWriter.close();

scanner.close();

}

}