Botas

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 62.5MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Botas**

Un lindo gatito se encontraba ordenando sus cosas. Entre sus pertenencias tenia espadas, sombreros, tazones, ratones de juguete y una colección de botellas de leche. Pero su tesoro mas preciado era una colección de botas. Como tenia un desorden total de las botas, quería reordenarlas y deshacerse de todas las botas que habían quedado huerfanitas. El gato tenia una gran colección, por que podría tener tantas botas como puedas imaginar.

**Problema**

Realiza un programa que dadas la cantidad de botas que tenia, y un identificador de cada bota, decir de cuantas botas se puede deshacer el gato, es decir, todas las que han perdido su par. Dos botas son del mismo par si tienen el mismo identificador.

**Entrada**

En la primer linea, un numero N con la cantidad de botas que tiene el gato. En la segunda linea, N números, el identificador de cada una de las botas

**Salida**

Un solo numero, con la cantidad de botas que puede tirar el gato.

**Ejemplo**

|  |  |
| --- | --- |
| Entrada | Salida |
| 10  2 4 3 2 2 3 5 5 8 10 | 4 |

**Límites**

* Para el 30% de los casos: 1<N<100
* Para el 50% de los casos: 1<N<1,000
* Para el 100% de los casos: 1<N<100,000

**Notas**

* Ninguna bota tendrá un identificador mayor a 106106.
* Las botas para gato son iguales tanto la izquierda como la derecha.

*Fuente: Concurso de ProgramaciÃ³n UVM 2012*

Problema subido por: [Gilberto Vargas](https://omegaup.com/profile/tacho.cpp/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/botas#problems/new-run) | | | | | | | | |
| 22db4d46 | Respuesta correcta | 100.00% | 0 | cpp | 6.32 | 0.36 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/botas#problems>

#include <iostream>

#include <stdio.h>

#include <map>

using namespace std;

int main() {

int N ;

scanf("%d", &N);

int arr[N];

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

scanf("%d", &arr[i]);

}

std::map<int,int> m;

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

m[arr[i]]++;

}

int answer = 0;

for(std::map<int,int>::iterator it = m.begin(); it != m.end(); it++) {

if(it->second % 2 != 0) {

answer++;

}

}

printf("%d", answer);

system("pause");

return 0;

}