Encontrando la posición.

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Descripción**

Dado un conjunto de 1<=N<=100 números enteros, determina en que lugar (ordinal) queda el número K después de ordenar el conjunto en forma creciente. Puedes estar seguro de que el entero K solo aparecerá una vez en el conjunto.

**Entrada**

En la primera línea el entero N. En la segunda línea y separados por espacios, los N datos del conjunto. en la tercera líne el entero K.

**Salida**

El entero que denota en que posición queda el entero K, después de ordenar el conjunto. Considera que las posiciones se cuentan a partir de 0.

**Ejemplos**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| 7  5 6 2 10 1 20 8  8 | 4 | Los datos del conjunto ordenados de forma creciente:  1 2 5 6 8 10 20  K=8 está en la posición 4. |

*Fuente: lacj20*

Problema subido por: [lacj20](https://omegaup.com/profile/lacj20/)

| **Enviado** | **GUID** | **Status** | **Porcentaje** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Encontrando-la-posicion#problems/new-run) | | | | | | | | |
| 2016-11-02 04:20:37 | c306f727 | Respuesta correcta | 100.00% | cpp | 3.08 MB | 0.01 s |  |  |
| 2016-11-02 04:17:18 | 19a1805f | Error de compilación | — | cpp | — | — |  |  |
| 2016-11-02 04:06:36 | 0749d2bf | Error de ejecución | 80.00% | java | 17.01 MB | 1.30 s |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Encontrando-la-posicion#problems>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <algorithm>

#define ll long long int

using namespace std;

int main() {

int n;

scanf("%d", &n);

std::vector<int> v;

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

int elem;

scanf("%d", &elem);

v.push\_back(elem);

}

std::sort(v.begin(), v.end());

int k;

scanf("%d", &k);

std::vector<int>::iterator it = find(v.begin(),v.end(), k);

int pos = std::distance(v.begin(), it);

printf("%d\n", pos);

system("pause");

return 0;

}

------------------MAS SIMPLE------------------

#include <iostream>

#include <stdio.h>

#include <vector>

#include <algorithm>

#define ll long long int

using namespace std;

int main() {

int n;

scanf("%d", &n);

int a[n];

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

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

}

int k;

scanf("%d", &k);

int ant =0;

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

if(a[i] < k) { //ME FIJO CUALES SON LOS MENORES A k, por lo tanto, son lo //s que van a ir antes de k

ant++;

}

}

printf("%d\n", ant);

//system("pause");

return 0;

}