

Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática

Grupo (C o F): _____

Nombres: _____

Laboratorio: _____ Puesto: _____ Usuario de Exacrc: _____

Normas de realización de la misión

1. Debes programar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc.domjudge/team>.
2. Escribe comentarios que expliquen tu solución, justifiquen por qué se ha hecho así y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribid vuestro **nombre y apellidos** en un comentario en la primera línea de cada fichero que subais al juez.
5. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.

1. Se tienen dos vectores de enteros ordenados y distintos entre sí a y b con n y $n - 1$ elementos respectivamente. Los elementos de b son los mismos que tiene a excepto uno que falta. Se pide implementar un algoritmo recursivo eficiente que encuentre ese valor que falta. Se debe indicar y justificar (escribiendo la recurrencia) el coste asintótico en el caso peor del algoritmo.

Entrada

La entrada comienza con el número de casos a tratar. Cada caso consta de 3 líneas, en la primera se da el número de elementos del primer vector, en la segunda los elementos del primer vector y en la tercera los elementos del segundo vector. Los valores de los vectores son números enteros.

Salida

Para cada caso de prueba se escribe en una línea el número que falta.

Entrada de ejemplo

```
5
5
10 20 30 40 50
10 30 40 50
5
10 20 30 40 50
10 20 30 40
5
10 20 30 40 50
20 30 40 50
2
1 8
1
1
6
```

Salida de ejemplo

```
20
50
10
8
6
```

2. Dada una secuencia no vacía de n números enteros ordenados de manera no decreciente (puede haber elementos repetidos) y dado un entero x , se pide encontrar en la secuencia el entero *más cercano* a x (en términos de distancia, es decir, del valor absoluto de la diferencia). En caso de empate se debe devolver el menor. *Diseña e implementa* un algoritmo *recursivo* eficiente que resuelva el problema. Justifica el *coste* del algoritmo implementado.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el valor de x , a continuación n y finalmente la secuencia de enteros.

Salida

Por cada caso de prueba el programa escribirá una línea con el entero más cercano a x contenido en la secuencia, según explicado en el enunciado.

Entrada de ejemplo

```
4
5
2
1 9
2
2
1 9
-2
7
-5 -3 -1 0 1 3 5
2
10
-1 0 1 1 2 2 2 3 6 9
```

Salida de ejemplo

```
1
1
-3
2
```