

Fundamentos de Algoritmia
Grados en Ingeniería Informática
22 de Noviembre de 2019

Grupo : A

Nombres y nicks: _____

Laboratorio: _____ **Puesto:** _____ **Usuario del Juez:** _____

Carta y nick _____

Normas de realización de la misión

1. Debes programar soluciones para cada ejercicio, 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.

Implementa un algoritmo recursivo eficiente que dado un vector de números enteros positivos y consecutivos del que se han eliminado todos los elementos impares menos uno, encuentre cuál es ese número impar que ha quedado en el vector. Plantea la recurrencia correspondiente al coste de este algoritmo e indica a qué orden de complejidad pertenece la solución a la misma.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de 2 líneas. En la primera se indica el número de elementos del vector una vez eliminados los elementos impares. En la siguiente se indican los elementos del vector. El final de los casos se marca con el valor cero. El vector tiene al menos un elemento. Se garantiza que siempre existe un elemento impar en el vector.

Salida

Para cada caso de prueba se escribe en una línea el elemento impar del vector.

Entrada de ejemplo

```
4
6 8 9 10
4
1 2 4 6
4
4 5 6 8
4
4 6 8 9
5
1 2 4 6 8
5
2 3 4 6 8
1
3
2
5 6
2
6 7
0
```

Salida de ejemplo

```
9
1
5
9
1
3
3
5
7
```