

Separable

Diseña un algoritmo iterativo de coste lineal que decida si un vector booleano es *separable* a partir de la siguiente especificación:

```
{long(v) ≥ n ≥ 0}  
fun separable(bool v[], int n) return bool r  
{r = ∃p : 0 ≤ p ≤ n : (∀i : 0 ≤ i < p : v[i]) ∧ (∀j : p ≤ j < n : ¬v[j])}
```

Entrada

La entrada consta de una serie de casos de prueba. Cada uno consta de dos líneas; la primera contiene el número de elementos de la secuencia ($n \leq 1000$) y la segunda los valores booleanos que componen la secuencia. La entrada termina cuando la longitud de la secuencia es un número negativo.

Ten en cuenta que en C++, por defecto, los valores booleanos *true* y *false* se escriben y leen como 1 y 0 automáticamente. Es decir, puedes leer los valores de la secuencia directamente como valores booleanos.

Salida

Para cada caso de prueba se mostrará una línea que contenga un 1 si el caso de prueba es *separable* y un 0 si no lo es. Mira el ejemplo de salida para más detalles.

Entrada de ejemplo

```
5  
1 1 1 0 0  
6  
1 1 1 1 0 1  
3  
0 0 1  
3  
0 0 0  
0  
-1
```

Salida de ejemplo

```
1  
0  
0  
1  
1
```

Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2014/2015 (prof. Antonio Sánchez Ruiz-Granados). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.