

# Creciente por los pelos y divertido

Teniendo en cuenta las dos definiciones siguientes:

- Se dice que un vector es *creciente por los pelos* cuando, además de ser creciente, la diferencia entre un elemento y el elemento siguiente es como mucho uno.

Por ejemplo, los siguientes vectores de tamaño 4 cumplen la definición:

1	2	3	4
---	---	---	---

1	2	2	3
---	---	---	---

1	1	1	1
---	---	---	---

Fíjate que al no exigirse que el vector sea *estrictamente creciente*, el último vector con todos 1's cumple la definición.

Por su parte el vector

1	2	1	2
---	---	---	---

no es creciente por los pelos, pues no es creciente.

- Los vectores aburridos son aquellos en los que hay elementos que aparecen repetidos muchas veces. Se dice que un vector es *d-divertido* cuando ningún elemento se repite más de  $d$  veces.

Ejemplos de vectores *1-divertido* son

1	2	3	4
---	---	---	---

4	3	7	0
---	---	---	---

pues ningún elemento aparece más de una vez (fíjate que, en realidad, los vectores anteriores son también *10-divertidos*, pues ningún elemento aparece más de 10 veces).

Especifica, diseña e implementa una función que reciba un vector de enteros de longitud  $0 \leq n \leq 1000$  y un parámetro  $d > 0$  y devuelva si el vector es *creciente por los pelos* y *d-divertido*. Escribe el invariante y función de cota que permitan demostrar la corrección del algoritmo implementado. Por último, justifica el coste del algoritmo conseguido.

## Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba registra en el primer número el número máximo de veces que se puede repetir un elemento en el vector, en el segundo, el número de elementos del vector, y a continuación los elementos del vector separados por blancos.

## Salida

Para cada caso de prueba se escribirá en una línea el vector modificado y en otra línea la posición de separación.

### Entrada de ejemplo

```
12
1 4 1 2 3 4
1 4 1 2 1 2
1 4 1 2 3 3
1 4 4 3 2 1
2 4 1 2 3 4
2 4 1 2 3 3
2 4 1 1 2 2
2 4 1 1 3 3
2 4 1 1 1 2
2 4 1 1 1 3
5 4 1 1 1 1
5 4 1 1 3 4
```

### Salida de ejemplo

```
SI
NO
NO
NO
SI
SI
SI
NO
NO
NO
SI
NO
```