

# ¡Están vivos!

Daniel Nudos, un importante investigador de la Universidad, ha realizado un extraño descubrimiento durante sus investigaciones sobre nuevos algoritmos. En un vector de enteros relleno de ceros y abandonado durante varios días ha empezado a crecer una nueva fauna que parece alimentarse de dichos ceros. Analizándolos en detalle ha podido descubrir que pasan por varias fases durante su vida: la niñez (fase '1'), la edad adulta (fase '2') y adulto con hijo (fase '3'). También ha podido descubrir que, en cada iteración de la vida, dependiendo de las fases de sus vecinos, pueden evolucionar, siempre respetando el orden de los índices del vector, es decir, de izquierda a derecha. Las reglas que siguen son las siguientes :

1. Si un niño (1) tiene alimento (0) a su lado (a su izquierda o su derecha), pasa a estado adulto (2). En caso contrario, muere (0).
2. Si dos adultos adyacentes (22) tienen alimento (0) en su casilla izquierda, en dicha casilla nace un niño (1) y ellos se convierten en adultos con hijos (33).
3. Si dos adultos adyacentes (22) no tienen alimento en su casilla izquierda pero sí tienen en su casilla derecha, en dicha casilla nace un niño (1) y ellos se convierten en adultos con hijos (33).
4. Si un adulto con hijos (3) tiene comida (0) a una distancia de 1 o 2 casillas, pasa a adulto (2), pero si tiene la comida más lejos de 2 casillas, entonces muere (0).

Escribe un programa que simule dicha vida y permita saber cómo evolucionará un estado inicial determinado en un plazo de N iteraciones.

Se debe tener en cuenta que fuera de los límites del vector no hay ni alimento ni individuos.

## Entrada

La entrada constará de 2 líneas. En la primera línea estará el tamaño del vector (N) y el número de iteraciones a realizar (It) y en la segunda línea, N números representando el estado inicial del vector.

$1 \leq N \leq 100$

$1 \leq It \leq 100$

## Salida

It+1 líneas de N números representando los estados del vector en cada iteración de la vida, incluyendo el estado inicial.

## Puntuación

**Test 1** (10 puntos):  $N \geq 4$ .  $It = 1$ . Solo se aplica la **regla 1**. A lo largo de toda la simulación, siempre quedan al menos 2 ceros al principio del vector y 2 ceros al final del vector.



Olimpiada Informàtica  
Comunitat Valenciana

## II Olimpiada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior  
Universitat de València - Escola Tècnica Superior d'Enginyeria

**Test 2** (30 puntos):  $N \geq 4$ . A lo largo de toda la simulación, siempre quedan al menos 2 ceros al principio del vector y 2 ceros al final del vector.

**Test 3** (60 puntos): Sin restricciones.

### Ejemplos

Entrada

```
6 1
0 0 1 2 0 0
```

Salida

```
0 0 1 2 0 0
0 0 2 2 0 0
```

Entrada

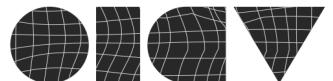
```
10 6
0 0 0 0 0 2 2 0 0 0
```

Salida

```
0 0 0 0 0 2 2 0 0 0
0 0 0 0 1 3 3 0 0 0
0 0 0 0 2 2 2 0 0 0
0 0 0 1 3 3 2 0 0 0
0 0 0 2 2 2 2 0 0 0
0 0 1 3 3 3 3 1 0 0
0 0 2 2 0 0 2 2 0 0
```

Entrada

```
12 5
2 2 0 0 1 3 3 1 2 2 2 0
```



Olimpiada Informàtica  
Comunitat Valenciana

## II Olimpiada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior  
Universitat de València - Escola Tècnica Superior d'Enginyeria

### Salida

```
2 2 0 0 1 3 3 1 2 2 2 0
3 3 1 0 2 2 0 0 2 3 3 1
0 2 2 1 3 3 0 0 2 2 0 0
1 3 3 0 2 2 0 1 3 3 0 0
0 2 2 1 3 3 0 2 2 2 0 0
1 3 3 0 2 2 1 3 3 2 0 0
```