

Arreglando el algoritmo

Tiempo máximo: 1,000-4,000 s Memoria máxima: 4096 KiB
<http://www.aceptaelreto.com/problem/statement.php?id=778>

Paula está empezando a estudiar algoritmia. Después de aprender la sintaxis del lenguaje toca ahora aprender los algoritmos más conocidos. Hoy le ha llegado el turno a los distintos métodos de ordenación entre los que se encuentra el *algoritmo de selección*.

La idea del algoritmo es sencilla. Si queremos ordenar de menor a mayor un vector v de n elementos, hacemos un primer recorrido del vector completo buscando el elemento más pequeño y lo colocamos en $v[0]$. Después volvemos a recorrer el vector, esta vez empezando desde la posición 1, para encontrar el elemento mínimo de nuevo y lo colocamos en $v[1]$. El proceso continúa iterativamente de forma que en un determinado momento tenemos los i primeros números ya colocados (en las posiciones $0 \dots i-1$), buscamos el elemento más pequeño en las posiciones $i \dots n-1$ y lo colocamos en $v[i]$.

El código en un lenguaje con sintaxis tipo C es algo así:

```
// v: vector a ordenar
// n: número de elementos
for (int i = 0; i < n; ++i) {
    int posMin = i;
    for (int j = i + 1; j < n; ++j)
        if (v[j] < v[posMin])
            posMin = j;
    swap(v[posMin], v[i]); // Intercambiamos los valores
}
```

Para practicar, Paula ha implementado una versión alternativa del algoritmo de selección que en lugar de ir ordenando el vector de izquierda a derecha buscando el mínimo cada vez, ordena de derecha a izquierda buscando el máximo. Lamentablemente ha fallado algo, porque al ejecutarlo ha saltado una excepción extraña y el algoritmo ha terminado de forma abrupta. Para buscar el error ha conseguido ver cómo estaba el vector en el momento del fallo y se pregunta cuántos valores había ordenado hasta ese momento.

Entrada

La entrada estará compuesta de varios casos de prueba. Cada caso consistirá en dos líneas. La primera contiene el número n de elementos que tiene el vector que Paula estaba intentando ordenar. La segunda línea contiene los valores del vector en el momento del fallo separados por espacios. Los valores son números enteros cuyo valor absoluto no supera 10^{18} .

Tras el último caso de prueba aparece una línea con un 0 que no debe procesarse.

Salida

Por cada caso de prueba se escribirá una línea con un único número que indica cuántos elementos, como mucho, había ordenado el algoritmo antes de fallar.

Entrada de ejemplo

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

Salida de ejemplo

| |
|---|
| 4 |
| 4 |

Autor: Marco Antonio Gómez Martín.

Revisores: Pedro Pablo Gómez Martín y Alberto Verdejo.