

Problema 2

Implementar un algoritmo que verifique la siguiente especificación

```
predicate EstricCreciente (s : seq<int>)
ensures EstricCreciente(s) == forall u , w :: 0 <= u < w < |s| ==> s[u] < s[w]

method Problema02(v : array<int>)
  requires v != null
  requires forall k :: 0 <= k < v.Length ==> v[k] > 0
  requires EstricCreciente(v[..])

  ensures forall i::0<=i<old(v.Length) && old(v[i])%2 == 0 ==> old(v[i]) in v[..]
  ensures forall i::0<=i<old(v.Length) && old(v[i])%2 == 1 ==> !(old(v[i]) in v[..])
  ensures 0 <= v.Length <= old(v.Length)
  ensures forall i :: 0 <= i < v.Length ==> v[i] in old(v[..])
  ensures EstricCreciente(v[..])
  modifies v
```

Requisitos de implementación.

- Utilizar la plantilla `PlantillaJuezNumeroCasos.cpp` que se encuentra en el campus virtual en la pestaña laboratorio.
- Los datos de entrada deben almacenarse en un *vector* de la clase `vector`. La función `resuelveCaso` leerá los datos, llamará a la función `resolver` con el vector y escribirá el resultado cuando la función termine de ejecutarse. La función `resolver` recibe el vector y lo modifica.
- La función `resolver` debe tener coste lineal.
- La entrada la lees como si fuera por teclado, utilizando el `std::cin`. Las instrucciones

```
std::ifstream in("datos.txt");
auto cinbuf = std::cin.rdbuf(in.rdbuf()); //save old buf and redirect std::cin to casos.
```

del `main` redirigen la entrada de teclado a fichero. Por lo tanto los datos no se teclearán, sino que se darán en un fichero. Baja del campus virtual el fichero con los casos del ejemplo, y modifica el nombre del fichero que aparece en la instrucción `std::ifstream in("datos.txt");` al nombre que tenga tu fichero de prueba. Ejecuta el programa y las instrucciones `std::cin` leerán los datos del fichero automáticamente.

- El método `resize()` de la clase `vector` redimensiona un vector al tamaño que se le indique. Por ejemplo `v.resize(n)` modifica el tamaño (número de elementos significativos) del vector `v` a `n`.
- La profesora ha resuelto el problema (función `resolver`) con 6 asignaciones, una instrucción condicional y una instrucción de repetición.
- Dafny no verifica automáticamente el programa hay que ayudarle por medio de `asserts` y `lemas` por lo que no lo verificaremos.

Entrada

La entrada comienza con un valor entero que indica el número de casos de prueba. Cada caso de prueba consta de dos líneas. La primera indica el número de elementos del vector. La segunda contiene los valores del vector.

El número de elementos del vector puede ser muy grande, por lo que no puede suponerse un valor máximo, y el valor de los elementos se sabe que pueden almacenarse en una variable de tipo `int`.

Salida

Para cada caso de prueba se escribe en una línea el vector resultado.

Entrada de ejemplo

```
3
8
3 4 6 7 11 14 16 18
3
3 5 7
4
2 4 6 8
```

Salida de ejemplo

```
4 6 14 16 18

2 4 6 8
```

Autor: Isabel Pita.