

Suma de valores.

Implementa una función que resuelva el siguiente problema:

```
function posMaximo (s : seq<int>) : nat
ensures 0 <= posMaximo(s) < |s|
ensures forall k :: 0 <= k < |s| ==> s[k] <= s[posMaximo(s)]

function SumSinMaximo(s:seq<int>):int
  ensures s==[] ==> SumSinMaximo(s)==0
  ensures s!=[] && posMaximo(s) != 0 ==>
    SumSinMaximo(s)==s[0]+SumSinMaximo(s[1..])
  ensures s!=[] && posMaximo(s) == 0 ==>
    SumSinMaximo(s)==SumSinMaximo(s[1..])

method SumaValores (v : array<int>) returns (s : int)
requires v != null && v.Length > 0
ensures s == SumSinMaximo(v[..])
```

Requisitos de implementación.

Indicar el coste de la solución obtenida. Solo se puede recorrer el vector una vez.

La función que resuelve el problema debe recibir como mínimo los datos en un vector y devolver el valor de la suma.

Entrada

La entrada comienza con el número de casos de prueba. Cada caso de prueba tiene dos líneas. En la primera línea se indica el tamaño del vector, en la segunda los valores separados por blancos.

Salida

Para cada caso de prueba se escribe en una línea el valor de la suma pedida.

Entrada de ejemplo

```
3
6
4 3 6 4 6 6
4
9 9 9 1
8
6 6 6 3 3 7 7 4
```

Salida de ejemplo

```
11
1
28
```

Autor: Isabel Pita.