

# Solo los buenos suman

Se desea implementar un algoritmo de coste lineal que satisfaga la siguiente especificación:

```
{0 ≤ N ≤ 20}
fun sumaBuenos(int v[N]) return int s
{s = (Σi : 0 ≤ i < N ∧ bueno(i, v) : v[i])}
```

donde  $bueno(i, v) \equiv (v[i] = 2^i)$ .

## Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación.

Cada caso de prueba se compone de dos líneas. La primera de ellas tiene un único entero con el número de elementos del vector (como mucho 20 elementos), mientras que la segunda línea contiene la lista con el contenido del vector (números enteros separados por espacios).

## Salida

Para cada caso de prueba aparecerá una línea con el resultado de la llamada a `sumaBuenos` para el vector correspondiente.

## Entrada de ejemplo

```
5
2
1 -2
4
1 2 4 8
5
0 0 0 0 16
0
4
2 4 8 16
```

## Salida de ejemplo

```
1
15
16
0
0
```

## Nota

Escribe como comentario en el código el invariante del bucle de tu algoritmo y la función *cota*.