

JS-字符串-在一条笔直的公路上安装了  $N$  个路灯

题目描述:

在一条笔直的公路上安装了  $N$  个路灯, 从位置 0 开始安装, 路灯之间间距固定为 100 米。  
每个路灯都有自己的照明半径, 请计算第一个路灯和最后一个路灯之间, 无法照明的区间的长度和。

输入描述:

第一行为一个数  $N$ , 表示路灯个数,  $1 \leq N \leq 100000$

第二行为  $N$  个空格分隔的数, 表示路灯的照明半径,  $1 \leq \text{照明半径} \leq 100000 * 100$

输出描述:

第一个路灯和最后一个路灯之间, 无法照明的区间的长度和

补充说明:

例 1

输入:

2

50 50

输出:

0

说明:

路灯 1 覆盖 0-50, 路灯 2 覆盖 50-100, 路灯 1 和路灯 2 之间(0 米-100 米)无未覆盖的区间

示例 2

输入:

4

50 70 20 70

输出:

20

说明:

[170,180],[220,230], 两个未覆盖的区间, 总里程为 20

```
const rl = require("readline").createInterface({ input: process.stdin });
```

```
var iter = rl[Symbol.asyncIterator]();
```

```
const readline = async () => (await iter.next()).value;
```

```
void async function () {
```

```
    // Write your code here
```

```
    let count = await readline();
```

```
    let lights = (await readline()).split(" ");
```

```
    let arr = Array(count);
```

```
    let res = 0;
```

```
    for(let i = 0; i < lights.length; i++) {
```

```
        arr[i] = [];
```

```
        let left = i > 0 ? 100 * i - lights[i] : 0;
```

```
        let right = 100 * i + Number(lights[i]);
```

```
        arr[i].push(left, right)
```

```
}  
arr.sort((a, b) => a[0] - b[0])  
for(let i = 0; i < arr.length - 1; i++) {  
  let curr = arr[i];  
  let next = arr[i+1];  
  if(curr[1] < next[0]) {  
    res += next[0] - curr[1];  
  } else {  
    next[1] = Math.max(curr[1], next[1])  
  }  
}  
  
// console.log(lights, arr)  
console.log(res)  
}()
```