Week 5: Assignment 5 - Question 2

BlockSum of an Array

Given an integer array M having size n which is power of 2,
Write a recursive code to find the BlockSum of the array M.

The following is the recursive definition of BlockSum:

If size of M is 2, say $M = [\ a,\ b\ ]$, where a and b are integers, then $BlockSum(M) = a - b$.

Otherwise (when n > 2), partition M into two subarrays of equal size:

$$M = [\ A\ B\ ]$$

The BlockSum of M is defined recursively as :

$$BlockSum(M) = BlockSum(A) - BlockSum(B)$$

Here A and B are arrays of Size n/2 each.
A is the first n/2 elements of M (in the same order) and B is the last n/2 elements of M (in the same order).

Note : You can assume that size of input array is a power of 2, and the size is less than 1024.

Input

The first line contains the array size n
The next n lines contains the elements of the array M.

Output

$BlockSum(M)$

Sample Inputs and Outputs

Sample Input 1

```
2
3 2
```

Sample Output 1

```
1
```

Sample Input 2

```
2
7 1
```

Sample Output 2

```
6
```

Sample Input 3

```
4
7 1 3 2
```

Sample Output 3

```
5
```

Explanation

BlockSum([3, 2]) = 3 - 2 = 1

BlockSum([7, 1]) = 7 - 1 = 6

BlockSum([7 , 1 , 3, 2]) = BlockSum([7, 1]) - BlockSum([3, 2]) = 6 - 1 = 5