

题目描述

A、B两个人把苹果分为两堆，A希望按照他的计算规则等分苹果，他的计算规则是按照二进制加法计算，并且不计算进位 $12+5=9$ ($1100 + 0101 = 9$)，B的计算规则是十进制加法，包括正常进位，B希望在满足A的情况下获取苹果重量最多。

输入苹果的数量和每个苹果重量，输出满足A的情况下B获取的苹果总重量。

如果无法满足A的要求，输出-1。

数据范围

$1 \leq \text{总苹果数量} \leq 20000$

$1 \leq \text{每个苹果重量} \leq 10000$

输入描述

输入第一行是苹果数量：3

输入第二行是每个苹果重量：3 5 6

输出描述

输出第一行是B获取的苹果总重量：11

用例

输入	3 3 5 6
输出	11
说明	无

输入	8 7258 6579 2602 6716 3050 3564 5396 1773
输出	35165
说明	无

题目解析

本题就是

题解可以[看链接博客](#)

JavaScript算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  const lines = [];
10 rl.on("line", (line) => {
11   lines.push(line);
12
13   if (lines.length === 2) {
14     const n = parseInt(lines[0]);
15     const arr = lines[1].split(" ").map(Number);
16
17     console.log(getResult(n, arr));
18
19     lines.length = 0;
20   }
21 });
22
23 function getResult(n, arr) {
24   const min = arr.sort((a, b) => a - b).shift();
25
26   let fault, correct;
27   fault = correct = min;
28
29   for (let w of arr) {
30     fault ^= w;
31     correct += w;
32   }
33
34   if (fault === 0) {
35     return correct - min;
36   } else {
37     return -1;
38   }
39 }
```

Java算法源码

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int n = sc.nextInt();
9         int[] arr = new int[n];
10        for (int i = 0; i < n; i++) {
11            arr[i] = sc.nextInt();
12        }
13
14        System.out.println(getResult(n, arr));
15    }
16
17    public static int getResult(int n, int[] arr) {
18        Arrays.sort(arr);
19
20        int min = arr[0];
21        int fault = min, correct = min;
22
23        for (int i = 1; i < arr.length; i++) {
24            int w = arr[i];
25            fault ^= w;
26            correct += w;
27        }
28
29        if (fault == 0) {
30            return correct - min;
31        } else {
32            return -1;
33        }
34    }
35 }
```

Python算法源码

```
1  # 输入获取
2  n = int(input())
3  arr = list(map(int, input().split()))
4
5
6  # 算法入口
7  def getResult():
8      arr.sort()
9      fault = correct = minV = arr[0]
10
11     for i in range(1, len(arr)):
12         w = arr[i]
13         fault ^= w
14         correct += w
15
16     if fault == 0:
17         return correct - minV
18     else:
19         return -1
20
21
22 # 算法调用
23 print(getResult())
```