

分苹果

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

补充说明：

按照 A 的计算方法 $5+6=3$ （ $101+110$ 不算进位的话值为 3）

示例

示例 1

输入：

3

3 5 6

输出：

11

说明：

示例 2

输入：

8

7258 6579 2602 6716 3050 3564 5396 1773

输出:

35165

说明:

```
import java.util.Scanner;
import java.util.*;
public class Main {
public static void main(String[] args) {
    /*Scanner      cin      =      new      Scanner(System.in,
StandardCharsets.UTF_8.name());
    int num = cin.nextInt();
    int[] steps = new int[num];
    for (int i = 0; i < num; i++) {
        steps[i] = cin.nextInt();
    }
    cin.close();*/
    Scanner sc = new Scanner(System.in);
    int res = -1;
    int n = sc.nextInt();
    int[] set = new int[n];
    int sum = 0;
    Arrays.sort(set);

    for (int i = 0; i < n; i++) {
        set[i] = sc.nextInt();
        sum += set[i];
    }

    Arrays.sort(set);

    int ans = 0;

    for (int i = 0; i < set.length; i++) {
        ans = ans ^ set[i];
    }

    if(ans != 0) {
        System.out.println(-1);
    } else {
        sum = sum - set[0];
    }
}
```

```
        System.out.println(sum);
    }
}
}
```