# Marc's Cakewalk

by satyaki3794



Marc loves cupcakes, but he also likes to stay fit. He eats $n$ cupcakes in one sitting, and each cupcake $i$ has a calorie count, $c_i$. After eating a cupcake with $c$ calories, he must walk *at least* $2^j \times c$ (where $j$ is the number cupcakes he has already eaten) miles to maintain his weight.

Given the individual calorie counts for each of the $n$ cupcakes, find and print a *long integer* denoting the minimum number of miles Marc must walk to maintain his weight. Note that he can eat the cupcakes *in any order*.

**Input Format**

The first line contains an integer, $n$, denoting the number of cupcakes.
The second line contains $n$ space-separated integers describing the respective calorie counts of each cupcake, $c_0, c_1, \ldots, c_{n-1}$.

**Constraints**

- $1 \leq n \leq 40$
- $1 \leq c_i \leq 1000$

**Output Format**

Print a long integer denoting the minimum number of miles Marc must walk to maintain his weight.

**Sample Input 0**

```
3
1 3 2
```

**Sample Output 0**

```
11
```

**Explanation 0**

Let's say the number of miles Marc must walk to maintain his weight is $miles$. He can minimize $miles$ by eating the $n = 3$ cupcakes in the following order:

1. Eat the cupcake with $c_1 = 3$ calories, so $miles = 0 + (3 \cdot 2^0) = 3$.
2. Eat the cupcake with $c_2 = 2$ calories, so $miles = 3 + (2 \cdot 2^1) = 7$.
3. Eat the cupcake with $c_0 = 1$ calories, so $miles = 7 + (1 \cdot 2^2) = 11$.

We then print the final value of $miles$, which is $11$, as our answer.

**Difficulty:** Easy

**Rate This Challenge:**

More

**Current Buffer** (saved locally, editable)

Java 8

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11         int n = in.nextInt();
12         long miles=0;
13         int[] calories = new int[n];
14         for(int calories_i=0; calories_i < n; calories_i++){
15             calories[calories_i] = in.nextInt();
16         }
17           for()
18     }
19 }
20
```

Line: 17 Col: 11

Upload Code as File  Test against custom input  Run Code  Submit Code