https://course.acciojob.com/idle?question=20d07877-efcd-4565-83b9-fc63961dec0d

- MEDIUM
- Max Score: 40 Points

Marc's Cakewalk

Marc loves cupcakes, but he also likes to stay fit. Each cupcake has a calorie count, and Marc can walk a distance to burn those calories.

On eating the j^{th} cupcake with `c` calories, he must walk at least `(2^j)*c` miles to maintain his weight.

For example, let calorie=[5,10,7]

If he eats the cupcakes in the order shown, the miles he will need to walk are $(2^0.5)+(2^1.10)+(2^2.7)=5+20+28=53$. This is not the minimum, though, so we need to test other orders of consumption.

In this case, our minimum miles is calculated as $(2^0.10)+(2^1.7)+(2^2.5)=10+14+20=44$.

Given the individual calorie counts for each of the cupcakes, determine 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, the number of cupcakes.

The next line inputs the calories of n cupcakes.

Output Format

Print the minimum miles necessary.

Example1

Input

```
3
1 3 2
```

Output

```
11
```

Explanation

Let us 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:

Eat the cupcake with $c_2=3$ calories, so $miles=0+(3.2^0)=3$.
Eat the cupcake with $c_1=2$ calories, so $miles=3+(2.2^1)=7$.
Eat the cupcake with $c_0=1$ calories, so $miles=7+(1.2^2)=11$.
We then print the final value of miles, which is 11, as our answer.

Constraints

1 <= n <= 40

1 <= c[i] <= 1000

Topic Tags

- **Arrays**

My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;
```

```
public class Main
{
        public static void main (String[] args) throws
java.lang.Exception
        {
                Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int c[] = new int[n];
      int temp;
      long sum=0;

      for(int i=0; i<n; i++){
        c[i] = sc.nextInt();
      }
      // sorting the array in order first
     for(int i=0; i<n; i++){
       for(int j=0; j<n; j++){
         if(c[i]>c[j]){
           temp =c[i];
           c[i]=c[j];
           c[j]=temp;
         }
       }
     }
      for(int i=0; i<n; i++){
        sum+=(c[i]*(Math.pow(2,i)));
      }
      System.out.print(sum);
        }
}
```