

Breaking Sticks

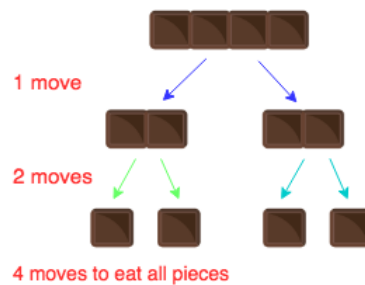


You recently received a bag of chocolate sticks for Halloween. To prevent you from compulsively eating all the chocolate sticks in one go, your dietician devises the following fun game.

In each move, you choose one of the sticks from your bag. Then, you either eat it, or break it into some number of equally-sized parts and save the pieces for later. The lengths of all sticks must always be integers, so breaking a stick into d parts is possible only if d is a divisor of the stick's length, and $d > 1$.

Note that this means that a stick of length 1 cannot be broken anymore, and can only be eaten.

For example, a chocolate stick of length 4 will be dealt with as shown below.



Complete the function *longestSequence* which takes, n , denoting the number of chocolate sticks and integer array a , denoting the lengths of the chocolate sticks, as input. Return the maximum, most optimal number of moves you can perform to consume the chocolate sticks according the game.

Given the chocolate sticks you received, determine the length of the longest sequence of moves you can perform.

Input Format

The first line contains one integer n which is the number of chocolate sticks.

The second line contains n space-separated integers a_1, a_2, \dots, a_n , the lengths of the chocolate sticks.

Constraints

- $1 \leq n \leq 100$
- $1 \leq a_i \leq 10^{12}$

Subtasks

- For 20% of the total score, $a_i \leq 10^6$

Output Format

Print a single integer which is the maximum number of moves you can perform.

Sample Input 0

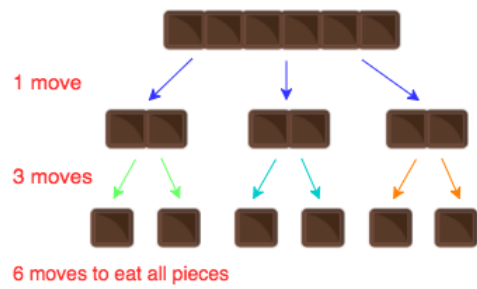
```
1
6
```

Sample Output 0

```
10
```

Explanation 0

You break a chocolate stick into three equal parts (1 move), then break each of them in half (3 moves), and then eat all six sticks (6 moves). This gives you 10 moves.



Sample Input 1

3
1 7 24

Sample Output 1

55

Explanation 1

You make 1 move using the stick of length 1, then 8 moves using the stick of length 7, and in the end 46 moves using the stick of length 24. This gives 55 moves in total.