```c
#include <stdio.h>

// Euclidean algorithm for GCD
int getGCD(int a, int b){
    if(a < b)
        return getGCD(b, a);
    if(a % b == 0)
        return b;
    return getGCD(b, a % b);
}

// a*b = LCM(a,b) * GCD(a,b)
int getLCM(int a, int b){
    return (a * b)/getGCD(a, b);
}

// n tells us how many elements are there in the array
// Actually it tells us how many elements we need to consider
// The array can have more elements than n but we will not care about them
int getFinishTime(int* times, int n){
    int finish = 1; // We will compute a running LCM
    for(int i = 0; i < n; i++)
        finish = getLCM(finish, times[i]); // LCM(a,b,c) = LCM(a,(LCM(b,c)))
    return finish;
}
int main(){
    int n;
    scanf("%d", &n);
    int times[n];
    for(int i = 0; i < n; i++)
        scanf("%d", &times[i]);
    printf("%d\n%d", getFinishTime(times, 2), getFinishTime(times, n));
    return 0;
}
```