4. Minimum Sum

Given an array of integers, perform some number *k* of operations. Each operation consists of removing any element from the array, dividing it by 2 and inserting the ceiling of that result back into the array. Minimize the sum of the elements in the final array.

Example:

nums = [10, 20, 7]

k = 4

Pick	Pick/2	Ceiling	Result
Initial array			*[10, 20, 7]*
7	*3.5*	*4*	*[10, 20, 4]*
10	*5*	*5*	*[5, 20, 4]*
20	*10*	*10*	*[5, 10, 4]*
10	*5*	*5*	*[5, 5, 4]*

The sum of the final array is *5 + 5 + 4 = 14,* and that sum is minimal.

Function Description

Complete the function *minSum* in the editor below. The function must return an integer denoting the minimum sum of the array after *k* steps.

minSum has the following parameters:

int nums[n]: an array of integers, indexed *0* to *n-1*

int k: an integer

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq num[i] \leq 10^4$ (where $0 \leq i < n$)
- $1 \leq k \leq 10^7$

Input Format For Custom Testing

The first line contains an integer, *n*, denoting the number of elements in *nums*.

Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) contains an integer describing *nums[i]*.

The last line contains an integer, *k*, denoting the number of moves.

Sample Case 0

Sample Input For Custom Testing

```
STDIN    Function
-----    --------
1    →   nums[] size n = 1
2    →   nums = [2]
1    →   k = 1
```

Sample Output

```
1
```

Explanation

In the first operation, the number *2* is reduced to *1*.

Sample Case 1

Sample Input For Custom Testing

```
STDIN    Function
-----    --------
2    →   nums[] size n = 2
2    →   nums = [2, 3]
3
1    →   k = 1
```

Sample Output

```
4
```

Explanation

In the first operation, either of the numbers may be reduced.

- If the number *2* gets reduced to *1*, the sum of the array is *4*.
- If the number 3 gets reduced to *2* (*3* divided by *2* equals *1.5, ceil(1.5) = 2*), the sum of the array is *4*.

The minimum sum of the array after one operation is *4*.