

Problem G. Merge Sort On Average

Input file: `mergesort.in`
Output file: `mergesort.out`
Time limit: 10 seconds
Memory limit: 256 megabytes

Merge sort algorithm sorts an array of numbers in the following way. Suppose the array has n numbers. When $n = 1$, we do nothing. Otherwise, we split it into two halves: the first half contains the first $\frac{n}{2}$ (rounded down) elements, and the second half contains all remaining elements. Then, we recursively sort the halves. And finally, we merge the sorted halves.

Merging is done as follows: we compare the first elements of the halves. The smaller one is the smallest element of the entire array, so we write it to the output and drop it from its half. Then, we take the first elements of the (remaining) halves, and so on. As soon as one of the halves becomes empty, we don't need to do any further comparisons as we can just copy the remaining half to the output. For simplicity, we assume all elements are distinct.

How many comparisons will this algorithm make, on average, when sorting a random n -element permutation?

Input

The first line of the input file contains an integer t ($1 \leq t \leq 50000$) — the number of testcases. The next t lines contain one integer n each ($1 \leq n \leq 10^9$). All testcases in one input file are distinct.

Output

Output t lines, with one floating-point number per line — the average number of comparisons for the corresponding n . Your answer will be considered correct if it's within 10^{-9} relative or absolute error from the correct answer.

Examples

<code>mergesort.in</code>	<code>mergesort.out</code>
2	2.6666666667
3	7.1666666667
5	