Bogosort

Recently Johnny have learned bogosort sorting algorithm. He thought that it is too ineffective. So he decided to improve it. As you may know this algorithm shuffles the sequence randomly until it is sorted. Johnny decided that we don't need to shuffle the whole sequence every time. If after the last shuffle several first elements end up in the right places we will fix them and don't shuffle those elements furthermore. We will do the same for the last elements if they are in the right places. For example, if the initial sequence is (3, 5, 1, 6, 4, 2) and after one shuffle Johnny gets (1, 2, 5, 4, 3, 6) he will fix 1, 2 and 6 and proceed with sorting (5, 4, 3) using the same algorithm. Johnny hopes that this optimization will significantly improve the algorithm. Help him calculate the expected amount of shuffles for the improved algorithm to sort the sequence of the first n natural numbers given that no elements are in the right places initially.

Input

The first line of input file is number t - the number of test cases. Each of the following t lines hold single number n - the number of elements in the sequence.

Constraints

1 <= t <= 175
2 <= n <= 175

Output

For each test case output the expected amount of shuffles needed for the improved algorithm to sort the sequence of first n natural numbers in the form of irreducible fractions.

Example

Input:
3
2
6
10

Output:
2
1826/189
877318/35343