Random modulo n

Kubík went to buy a pizza. To his surprise, the pizza box was made out of recycled… punch cards!

With his eagle eye, he deciphered the program the punch cards described:

```
n = read_input();
ans = 0;
while(n > 0)
{
    ans = ans + 1;
    n = random() % n;
}
```

random() is a function which returns uniformly random non-negative integers, and % is the modulus operator.

Now he wonders what the expected value of *ans* would be for a given initial value of *n*, and he is unable to enjoy his pizza until someone computes the answer for him.

Input

The first line contains an integer **1 ≤ T ≤ 5** - the number of test cases.

Each of the next **T** lines contain a single integer ***n***, where **1 ≤ *n* ≤ 300 000**. The sum of **n** within an input file won't exceed 300000.

Output

Output the expected value of the variable ***ans*** – that is, the sum of ***v*** × (probability that ***ans*** will end up with value ***v***), for all possible values ***v***.

Your answer will be considered correct if the absolute or relative error does not exceed 10^{-9}. Make sure to print enough decimal places.

Example

Input:

```
2
2
47
```

Output:

```
1.5
4.4379638417
```

In the first case, either random()%2 = 0 with probability 1/2, which leads to ans = 1, or random()%2 = 1 with probability 1/2, after which we certainly get random()%1 = 0, so ans = 2.

Expected value of ans is therefore 1 × 1/2 + 2 × 1/2 = 1.5.