



Problem H. Sum of all permutations

Input: Standard
Output: Standard
Author(s): Sebastián Gómez - UTP Colombia

Toby is very bored because his father went to live to Brazil, so he decided to create a challenge that might take a lot of time to solve. First he creates a function called

SadToby

that receives an array of integers called permutation and a number M as follows:

```
def SadToby(permutation , M):  
    sum = 0  
    for each x in permutation:  
        if (x <= M):  
            sum = sum + x  
        else:  
            break  
    return sum
```

For every permutation of the numbers from 1 to N Toby needs to print the sum of SadToby function. Toby needs to compute this result for every possible value of M between 1 and N . As each of this values can be very large output the result modulo the prime $p = 1711276033 = 2^{25} \times 51 + 1$. Can you help this cute dog with his task?

Input

The input consists of several test cases. Each test case begins with a line with one integers N .

- $1 \leq N \leq 10^5$

Output

For each test case, print a single line with N integers containing the required sum for every value of M between 1 and N .

Example

Input	Output
1	1
2	1 6
3	2 9 36

Explication

Third case, first output number $M = 1$. Consider all permutations. If the first number is greater than 1, then the loop will break in the beginning itself with output 0. There are a total of 6 distinct permutations out of which 4 will give 0. The remaining 2 will fetch 1 each from the function. Thus the answer is 2. For $M = 2$ it's easy to check that the output is 9 and for $M = 3$ is 36.