T 115 Reset Buttons

Problem	Submissions	Leaderboard

Kushal is trying to open a rather challenging lock. The lock has n buttons on it and to open it, you should press the buttons in a certain order to open the lock. When you push some button, it either stays pressed into the lock (that means that you've guessed correctly and pushed the button that goes next in the sequence), or all pressed buttons return to the initial position. When all buttons are pressed into the lock at once, the lock opens.

Consider an example with three buttons. Let's say that the opening sequence is: {2, 3, 1}. If you first press buttons 1 or 3, the buttons unpress immediately. If you first press button 2, it stays pressed. If you press 1 after 2, all buttons unpress. If you press 3 after 2, buttons 3 and 2 stay pressed. As soon as you've got two pressed buttons, you only need to press button 1 to open the lock.

Kushal doesn't know the opening sequence. But he is really smart and he is going to act in the optimal way. Calculate the number of times he's got to push a button in order to open the lock in the worst-case scenario.

Input Format

A single line contains integer n (1 ≤ n ≤ 2000) — the number of buttons the lock has.

Constraints

1<=n<=2000

Output Format

In a single line print the number of times Kushal has to push a button in the worst-case scenario.
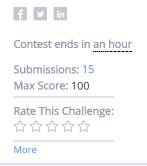
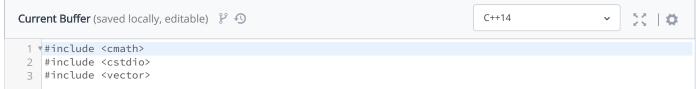## Sample Input 0

```
2
```

Sample Output 0

```
3
```

Explanation 0

Consider the test sample. Kushal can fail his first push and push the wrong button. In this case he will already be able to guess the right one with his second push. And his third push will push the second right button. Thus, in the worst-case scenario he will only need 3 pushes.

Contest ends in an hour

Submissions: 15
Max Score: 100

Rate This Challenge:
☆☆☆☆☆

More

Current Buffer (saved locally, editable)   C++14

```
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
```

```
 4  #include <iostream>
 5  #include <algorithm>
 6  using namespace std;
 7
 8
 9  int main() {
10      /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11      return 0;
12  }
13
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code