
DSA Lab 7 Set 1 | StackFact

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

When you try to find the factorial of a number using your recursive factorial function, you will see that you cannot go beyond even very small numbers. In this assignment you will be required to print out the factorial of a large number using the stack data structure.

Let us see how you can use a stack to add two large numbers. Let the first number be 9654321 and the other number be 87823. We read the first number digit by digit from the left and push each digit in stack *A* with top starting from value 9, and ending in top value being 1. Similarly, we read the second number and push it in stack *B* with top value being 3. Now we pop stack *A* to yield 1 and pop stack *B* to yield 3. We then add the two digits and push the result in a third stack *R*. We repeat the process till both stacks are empty. We may need to handle carryover and the case when only one stack is non-empty. Now if we pop stack *R*, we will be able to print the result in correct order.

You need to extend this concept for multiplying two numbers in order to compute factorial of a large number *N*.

Input

The input contains a single integer *N*.

Constraints:

Basic:

$$1 \leq N \leq 20$$

Advanced:

$$1 \leq N \leq 1000$$

Output

The output is a single integer denoting the value of *N!*.

Note

Sample Test Case:

Input:

25

Output:

15511210043330985984000000

Hint:

Consider multiplying $y*64$ for some number y , you should proceed with using stacks to perform multiplication $y*4$ and store the result in the stack(as explained for addition). Then you should perform $y*6$ and store the result in another stack. Figure out the logic to add the two results to obtain the final product. There is no restriction on the number of stacks.