Problem H. Shortest Path System

Input file:	`shortsystem.in`
Output file:	`shortsystem.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider an undirected graph. It has n vertices, and some edges, each connecting two distinct vertices and having a positive integer *length*. At most one edge connects each pair of vertices.

For each pair of vertices we can find the shortest path connecting them (the length of a path is the sum of lengths of edges in it). For some pairs of vertices, there might be several shortest paths connecting them. For some other pairs, there might be no path connecting them at all. We will not consider such graphs in this problem: we will only consider graphs where there is a path connecting any pair of vertices (the graph is *connected*) and where there is a unique shortest path connecting each pair of vertices.

We can now forget about the edges and their lengths, and just remember all shortest paths. We define a *shortest path system* as n vertices plus a set of $\frac{n(n-1)}{2}$ paths, one for each pair of vertices. Each path is simply a sequence of vertices. Of course, not any set of paths is a shortest path system — the system must still correspond to some graph with positive integer edge lengths.

How many different shortest path systems are there with n vertices? Two systems are considered the same if one can renumber the vertices in one system to obtain the other.

Input

The only line of the input file contains one integer n, $2 \le n \le 6$.

Output

Write the number of shortest path systems with n vertices to the output file.

Examples

shortsystem.in	shortsystem.out
3	2
4	6

Note

The two shortest path systems with 3 vertices are: the one where the shortest path between each pair of vertices is just an edge connecting those vertices, and the one where the shortest paths from some vertex to two others are just edges, whereas the shortest path between the others goes through the first vertex.