Green Day

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Consider a graph consisting of $n \geq 2$ vertices without loops or parallel edges. Each edge may be colored in one of k colors. We call a coloring *proper* if edges of each color form a spanning tree of the graph (that is, for each color c, there exists a unique path between each pair of vertices that uses only edges of color c). Denote such spanning tree for color c as T_c.

We call a *proper* coloring *safe* if for each two colors c and d and for each two distinct vertices u and v, the following statement is correct: $\mathrm{path}_{T_c}(u, v) \cap \mathrm{path}_{T_d}(u, v) = \{u, v\}$, where $\mathrm{path}_T(u, v)$ is the set of all vertices of tree T that lie on the simple path between u and v (including u and v themselves).

Your task is to construct such a graph that its edges are colored in k colors forming a *safe proper* coloring.

Input

The first and only line of input contains a single positive integer k ($2 \leq k \leq 100$), the number of colors you should use in your graph.

Output

On the first line, output $n \geq 2$: the number of vertices in your graph.

Then, output k groups consisting of $n - 1$ edges representing edges of each color. Output each edge as a pair of integers a_i, b_i on a separate line ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$).

Your output must satisfy the condition $(n - 1) \cdot k \leq 10^6$. There must be no parallel edges.

You are allowed to output any valid answer. It's guaranteed that at least one solution exists.

Example

standard input	standard output
2	4 1 2 1 3 3 4 4 1 2 3 2 4