

Distinct Orders

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

You regularly visit a taverna in Heraklion for meze, a selection of small dishes. To keep the dinners varied, the cook alternates between two recipes for each dish.

The menu offers n dishes, numbered from 1 to n . Each dish has two versions: the first and the second.

Each day, you choose an integer k from 1 to n and order one portion of each of the first k dishes: dishes $1, 2, \dots, k$. You cannot choose their versions. The first time you order a dish, you receive its first version. On every subsequent occasion, you receive the opposite version from the last time you ordered that dish. Days when you do not order a dish do not affect its version.

Two daily orders are identical if they contain the same number of dishes and the version of each corresponding dish is the same.

Construct a sequence $k_1, k_2, \dots, k_m$ of maximum possible length such that no two daily orders are identical.

Input

The only line contains an integer n ($1 \leq n \leq 20$), the number of dishes.

Output

In the first line, print an integer m , the maximum possible number of days.

In the second line, print m integers $k_1, k_2, \dots, k_m$ ($1 \leq k_i \leq n$), the number of dishes to order on each day.

If there are several optimal sequences, print any of them.

Examples

standard input	standard output
1	2 1 1
2	6 2 2 1 2 2 1

Note

In the first example, order the only dish twice. You receive its first version and then its second version.

In the second example, the sequence $2, 2, 1, 2, 2, 1$ produces the following orders. The numbers in parentheses are the versions of dishes $1, 2, \dots, k$:

$$(1, 1), \quad (2, 2), \quad (1), \quad (2, 1), \quad (1, 2), \quad (2).$$

All six orders are distinct.