

A. Save the problem!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Attention: we lost all the test cases for this problem, so instead of solving the problem, we need you to generate test cases. We're going to give you the answer, and you need to print a test case that produces the given answer. The original problem is in the following paragraph.

People don't use cash as often as they used to. Having a credit card solves some of the hassles of cash, such as having to receive change when you can't form the exact amount of money needed to purchase an item. Typically cashiers will give you as few coins as possible in change, but they don't have to. For example, if your change is 30 cents, a cashier could give you a 5 cent piece and a 25 cent piece, or they could give you three 10 cent pieces, or ten 1 cent pieces, two 5 cent pieces, and one 10 cent piece. Altogether there are 18 different ways to make 30 cents using only 1 cent pieces, 5 cent pieces, 10 cent pieces, and 25 cent pieces. Two ways are considered different if they contain a different number of at least one type of coin. Given the denominations of the coins and an amount of change to be made, how many different ways are there to make change?

As we mentioned before, we lost all the test cases for this problem, so we're actually going to give you the number of ways, and want you to produce a test case for which the number of ways is the given number. There could be many ways to achieve this (we guarantee there's always at least one), so you can print any, as long as it meets the constraints described below.

Input

Input will consist of a single integer A ($1 \leq A \leq 10^5$), the desired number of ways.

Output

In the first line print integers N and M ($1 \leq N \leq 10^6$, $1 \leq M \leq 10$), the amount of change to be made, and the number of denominations, respectively.

Then print M integers D_1, D_2, \dots, D_M ($1 \leq D_i \leq 10^6$), the denominations of the coins. All denominations must be distinct: for any $i \neq j$ we must have $D_i \neq D_j$.

If there are multiple tests, print any of them. You can print denominations in arbitrary order.

Examples

input
18
output
30 4 1 5 10 25

input
3
output
20 2 5 2

input
314
output

