G Git Gud



You are an adventurer working for the futuristic corporation RoboCorp. Your initial skill level is an integer in the range $[1, n]$, but you don't know its exact value. Your goal is to reach a skill level of at least n by completing missions for RoboCorp, but each mission costs precious robocoins.

You can choose missions of any difficulty and any positive integer duration (in hours). However, the cost depends both on the new mission's length, and on how the new mission's difficulty compares to your most recent one.

Let x be the difficulty of your most recent mission. If you want to undertake a new mission of difficulty y and duration l:

- If it is your first mission, or $y \leq x$, the cost is l robocoins.
- If $y > x$, the cost is $1000 + l$ robocoins.

Your skill improves only when you take on a mission whose difficulty y exactly matches your current skill s:

- If $y = s$, then your skill increases by l.
- Otherwise, your skill does not change.

After each mission, you still don't know your actual skill value.

You start with 10^6 robocoins. Find a strategy (a sequence of missions) that does not exceed your budget and guarantees your skill becomes at least n, no matter what your initial skill was.

Input

The input contains a single line with an integer n ($1 \leq n \leq 250\,000$) — the target skill level (that is, by the end, your skill must be greater or equal to n).

Output

In the first line, print a single integer k ($0 \leq k \leq 10^6$) — the number of missions you plan to complete.

Then, print k lines. The i-th line must contain two integers y and l ($1 \leq y, l \leq 10^6$) — the difficulty and duration (in hours) of the i-th mission, respectively.

Samples

Sample input 1	Sample output 1
4	4 1 4 3 1 2 1 3 1

Explanation of sample 1.
In the example, your target skill is $n = 4$. You can use the following strategy:

- Take a mission of difficulty $y = 1$ and duration $l = 4$. It's your first mission, so you pay $l = 4$ robocoins.
- Take a mission of difficulty $y = 3$ and duration $l = 1$. The previous mission had difficulty $x = 1$, so since $y > x$, you pay $l + 1000 = 1001$ robocoins.
- Take a mission of difficulty $y = 2$ and duration $l = 1$. Now $x = 3$, and since $y \leq x$, you pay $l = 1$ robocoin.
- Take a mission of difficulty $y = 3$ and duration $l = 1$. Now $x = 2$, and since $y > x$, you pay $l + 1000 = 1001$ robocoins.

In total, you spend 2007 robocoins, which is within your 10^6-robocoin budget.

Now we verify that this strategy always works:

- If your initial skill was 1, after the first mission it becomes 5.
- If your initial skill was 2, after the third mission it becomes 3, and after the fourth mission it becomes 4.
- If your initial skill was 3, after the second mission it becomes 4.
- If your initial skill was 4, it stays 4.

So no matter what your starting skill is, you end with skill $\geq n$.

Image credits: Evan-Amos, CC BY-SA 3.0, via Wikimedia Commons