UTS Open '24 P4 - Political Espionage

Time limit: 1.0s **Memory limit:** 512M

Marcus is trying to create yet another data structure problem! The problem statement is as follows:

> In the land of DMOJistan, citizens live in N cities along a number line, conveniently numbered from 1 to N. The i^{th} city is at position i and has a population of a_i.
>
> As a member of the PEG (political espionage group), you have been tasked with ensuring that one city holds a majority of the power. Specifically, you will readjust the borders of DMOJistan to only contain the cities in the range $[l, r]$. How many pairs (l, r) are there such that the city with the greatest population in the range $[l, r]$ has **strictly greater than half** the total population of the range?

Marcus attempted to create strong data for this problem, but Edward kept finding ways to cheese it. After Marcus gave up, Edward suggested that the data generation could be a problem in and of itself! Edward quickly finds an optimal construction, but Marcus has no idea what to do. Can you help him?

Your job is to construct an array of N integers $a_1, a_2, \ldots a_N$ in the range $[1, 10^{10}]$, such that the answer to the above problem statement is as large as possible. Your score will be calculated based on how close you get to the maximum achievable number of pairs.

Constraints

For all subtasks,
$1 \le N \le 2 \times 10^5$
$1 \le a_i \le 10^{10}$

K is a variable used for scoring. (More details in the scoring section).

Subtask 1 [30%]

$1 \le N \le 1000$
$K = 2$

Subtask 2 [70%]

$1 \le N \le 2 \times 10^5$
$K = N$

Input Specification

The first and only line of input contains one integer, N, the desired length of the array.

Output Specification

Output one line containing N space-separated integers $a_1, a_2, \ldots a_N$ in the range $[1, 10^{10}]$, your constructed array.

Scoring

If you do not output N integers in the range $[1, 10^{10}]$ or your output is formatted incorrectly, you will receive a Wrong Answer verdict.

Otherwise, let X be the number of pairs in your array. Let Y be the maximum number of pairs across all possible arrays. Your score will be calculated as

$$\left\lfloor \left(\frac{X}{Y}\right)^{K} \times 100\% \right\rfloor$$

where K is the scoring variable defined for each subtask. Your score for a subtask is the minimum of the scores of all of the tests within the subtask.

Sample Input

```
6
```

Sample Output

```
3 1 4 1 5 9
```

Explanation for Sample

There are 13 valid pairs (l, r) in this construction:

$(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)$
$(1,2), (2,3), (3,4), (4,5), (5,6)$
$(2,4), (4,6)$

A pair such as $(1,3)$ is not counted since $\max(3,1,4) = 4$ is not strictly greater than $\frac{1}{2}(3+1+4) = 4$.

It can be shown that this construction is not optimal.