Randomised Quick Sort Analysis 1

In this problem, we want to analyse the randomised version of quicksort.

The implementation that we will analyse is given here.

We will measure the time complexity by the number of **comparisons** that the sorting procedure makes.

In this problem we want to analyse the average number of comparisons made for a **given array**.

Given an integer n

, find an array of size n, such that the average number of comparisons performed by quicksort on this array is $n * (n-1)2$

.

Note that the maximum possible value of the number of comparisons is $n * (n-1)2$

, so if the average has to be equal to this value, the number of comparisons to be made must be equal to this value, **regardless** of the choice of pivot.

There may be multiple possible choices for the array A

, you can output **any** of them.

For preventing large files, the integers in the array should be between -109

and 109

(inclusive).

It is guaranteed that there exists at least one correct answer for all n

given as input.

This shows that there are examples where the number of comparisons of quick sort is bad, regardless of the how "random" the pivot is chosen!

Input

A single integer n

, the size of the array

Output

A single line with n

space separated integers, the elements of the array

For preventing large files, the integers should be between -109

and 109

(inclusive).

Judging

If there are less than n

space separated integers, in the given array or their values are not in the range specified, then you will receive a "Wrong Answer" verdict.

If the average number of comparisons performed by the given quicksort implementation on your array is equal to $n * (n-1)2$

, then the answer is considered correct, otherwise you will receive a "Wrong Answer" verdict.

Constraints

$1 \le n \le 500$

Sample Input

```
2
```

Sample Output

```
1 2
```

Explanation

Whatever be the choice of the pivot, there will be only one comparison made, therefore the average number of comparisons is $1=2 * (2-1)2$

.

Sample Input 2

```
3
```

Sample Output 2

```
1 2 1
```

Explanation

This output will give a wrong answer and is given only to explain the problem.

If the first pivot chosen is the index 0

, then 2 comparisons are made and the array becomes $\{1,1,2\}$, the subarrays $\{2\}$ and $\{1\}$

are recursively sorted and no more comparisons are made.

In this possiblity we have only 2

comparisons made, and so the average must be less than 3 (because the maximum number of comparisons in the worst case is 3

).

So this output is not correct.