

## Km to Go (kmtogo)

Karcsi is organizing a running race of  $N$  kilometers and wants to print large numeric signs on A4 sheets to indicate the remaining distance at each kilometer mark.



Figure 1: Ready, steady, go!

The signs display the numbers from  $N$  to 1, showing the number of kilometers left to the finish line. Each sheet contains exactly one digit, meaning that multi-digit numbers are split across multiple sheets.

Karcsi wants to know how many copies of each digit he must print for the race. Write a program to calculate how often each digit  $(0, 1, \dots, 9)$  appears amongst the numbers  $N, N-1, \dots, 2, 1$ .

 Among the attachments of this task you may find a template file `kmtogo.*` with a sample incomplete implementation.

### Input

A single number  $N$ , the length of the race.

### Output

You need to write a single line with ten integers  $D_0, D_1, \dots, D_9$ , where  $D_i$  represents how many times the digit  $i$  ( $0 \leq i \leq 9$ ) appears in the numbers (kilometers) from 1 to  $N$ .

### Constraints





- $1 \leq N \leq 1\,000\,000$ .

### Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points)      Examples.



- **Subtask 2** (5 points)  $N \leq 9$ .  

- **Subtask 3** (30 points)  $N \leq 99$ .  

- **Subtask 4** (20 points)  $N \leq 9999$ .  

- **Subtask 5** (45 points) No additional limitations.  


## Examples

input	output
12	1 5 2 1 1 1 1 1 1 1
9752	2845 3956 3956 3955 3955 3948 3945 3898 3845 3598

## Explanation

In the **first sample case** Karcsi has to print `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`. So he needs *one* `0`, *five* `1`, *two* `2`, ..., *one* `9` digit.