E2. Numerical Sequence (hard version)

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

The only difference between the easy and the hard versions is the maximum value of k.

You are given an *infinite* sequence of form "112123123412345..." which consist of blocks of all consecutive positive integers written one after another. The first block consists of all numbers from 1 to 1, the second one — from 1 to 2, the third one — from 1 to 3, ..., the i-th block consists of all numbers from 1 to i.

So the first 56 elements of the sequence are "11212312341234512345612345671234567812345678912345678910". Elements of the sequence are numbered from one. For example, the 1-st element of the sequence is 1, the 3-rd element of the sequence is 2, the 20-th element of the sequence is 5, the 38-th element is 2, the 56-th element of the sequence is 0.

Your task is to answer q independent queries. In the i-th query you are given one integer k_i. Calculate the digit at the position k_i of the sequence.

Input

The first line of the input contains one integer q ($1 \le q \le 500$) — the number of queries.

The i-th of the following q lines contains one integer k_i ($1 \le k_i \le 10^{18}$) — the description of the corresponding query.

Output

Print q lines. In the i-th line print one digit x_i ($0 \le x_i \le 9$) — the answer to the query i, i.e. x_i should be equal to the element at the position k_i of the sequence.

Examples

input

```
5
1
3
20
38
56
```

output

```
1
2
5
2
0
```

input

```
4
2132
506
999999999999999999
1000000000000000000
```

output

```
8
2
4
1
```

Note

Answers on queries from the first example are described in the problem statement.