

## Question 1

### Problem: Reverse the Matrix

You are given an  $n \times m$  matrix of integers. Write a program that rotates the matrix by 180 degrees.

---

#### Input

- The first line contains two integers  $n$  and  $m$  — the number of rows and columns of the matrix.
- The next  $n$  lines each contain  $m$  integers, representing the matrix.

#### Output

- Print the resulting  $n \times m$  matrix after rotating it 180 degrees.
  - Elements in each row should be space-separated.
- 

### Example 1

#### Input

3 3

1 2 3

4 5 6

7 8 9

#### Output

9 8 7

6 5 4

3 2 1

---

### Example 2

#### Input

2 4

10 20 30 40

50 60 70 80

#### Output

80 70 60 50

40 30 20 10

## Question 2

### Problem: Largest Left-Truncatable Happy Number Below N

A positive integer is called **happy** if the following process eventually reaches 1:

- Replace the number by the sum of the squares of its digits.
- Repeat the process until either the number becomes 1 (happy) or it cycles endlessly without reaching 1 (unhappy).

For example:

- 19 is happy because  $1^2 + 9^2 = 82$ , then  $8^2 + 2^2 = 68$ , then  $6^2 + 8^2 = 100$ , then  $1^2 + 0^2 + 0^2 = 1$ .
- 4 is unhappy because it cycles through  $4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4$  and never reaches 1.

 An unhappy number will **fall into a cycle** of repeated values forever. For example:

$2 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow \dots$

**Cycle reached:**  $4 \rightarrow 16 \rightarrow \dots$  (so 2 is unhappy)

$3 \rightarrow 9 \rightarrow 81 \rightarrow 65 \rightarrow 61 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \dots$

**Cycle reached:**  $37 \rightarrow 58 \rightarrow \dots$  (3 is unhappy)

$5 \rightarrow 25 \rightarrow 29 \rightarrow 85 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \dots$

**Cycle reached:**  $89 \rightarrow 145 \rightarrow \dots$  (5 is unhappy)

A number is called **left-truncatable happy** if it is happy **and** every number obtained by repeatedly removing digits from the left is also happy.

Example: 931 is left-truncatable happy because 931, 31, and 1 are all happy.

---

### Task

Given an integer  $n$ , find the largest integer **less than  $n$**  that is left-truncatable happy.

---

Program input and output will make use of stdio streams (System.in and System.out in Java) i.e., not file I/O.

### Input

- A single integer  $n$ ,  $n \geq 2$

### Output

- Print the largest happy, left-truncatable integer less than  $n$ .
- 

### Example 1

#### Input

20

#### Output

7

---

**Example 2****Input**

100

**Output**97

---

**Example 3****Input**

1234

**Output**

931