**1.** Write a function to print the following complex star pattern of size n:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Input Format**

The input consists of a single integer n (1 <= n <= 100), which represents the size of the pattern.

**Constraints**

The pattern size (n) will be a positive integer between 1 and 100.

**Output Format**

Print the complex star pattern of size n, as shown in the problem statement.

**Sample Input 0**

5

**Sample Output 0**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Explanation 0**

The pattern has 5 rows and columns, with the center row containing only one star. The stars are positioned in a way that creates a diamond shape.

**2.** Output the ID(s) of the most influential user(s) in the network, based on the number of connections they have. If there are multiple users with the same number of connections, output all their IDs in ascending order. You are given a list of user IDs and their connections with other users. Each connection is represented as an edge between two user IDs. Your task is to identify the most influential user(s) in the network, based on the number of connections they have.

**Input Format**

The input consists of two integers n and m, where n represents the number of users in the network and m represents the number of edges between the users. This is followed by m lines, each containing two integers u and v, which represent an edge between user u and user v.

**Constraints**

* 1 <= n <= 10^6
* 1 <= m <= 10^6
* 1 <= u, v <= n
* The input graph may not be connected.

**Output Format**

Output the ID(s) of the most influential user(s) in the network, based on the number of connections they have. If there are multiple users with the same number of connections, output all their IDs in ascending order.

**Sample Input 0**

6 5

1 2

1 3

2 3

3 4

5 6

**Sample Output 0**

3

**Explanation 0**

In this example, the user 3 has the most connections, with a total of 3 connections. Users 1, 2, 4, 5, and 6 each have 1 connection. Therefore, the output is 3.

**Sample Input 1**

10 9

1 2

1 3

1 4

2 3

2 4

2 5

3 4

3 6

4 7

**Sample Output 1**

2 3 4

**Explanation 1**

In this example, users 2, 3, and 4 have the most connections, each having 4 connections. Users 1 and 5 each have 1 connection, while users 6 and 7 each have 2 connections. Users 8, 9, and 10 are not connected to anyone else. Therefore, the output is 2 3 4.

**3.** You are given a string s of length n. Your task is to compress the string to the shortest possible length by replacing repeated characters with the number of occurrences of the character followed by the character itself. If the compressed string is not shorter than the original string, then return the original string.

For example, given the string "aabcccccaaa", the compressed string would be "a2b1c5a3". If the compressed string is not shorter than the original string, then return the original string.

**Input Format**

The input consists of a single string s, where 1 <= n <= 10^6.

**Constraints**

The input string s consists of lowercase English letters.

**Output Format**

Print the compressed string if it is shorter than the original string, otherwise print the original string.

**Sample Input 0**

aabcccccaaa

**Sample Output 0**

a2b1c5a3

**Explanation 0**

The compressed string is shorter than the original string, so we print the compressed string.

Note: In the compressed string, the number of occurrences of each character is represented as a single digit. If the number of occurrences is greater than 9, then the digits are split and represented separately. For example, "aaaaaaaaaaa" would be compressed as "a11" and not "a9a2".

**Sample Input 1**

abcd

**Sample Output 1**

abcd