Medium

Table: Employees

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| employee_id   | int     |
| employee_name | varchar |
| manager_id    | int     |
+---------------+---------+
```

employee_id is the column of unique values for this table.
Each row of this table indicates that the employee with ID employee_id and name employee_name reports his work to his/her direct manager with manager_id
The head of the company is the employee with employee_id = 1.

Write a solution to find employee_id of all employees that directly or indirectly report their work to the head of the company.

The indirect relation between managers **will not exceed three managers** as the company is small.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

Input:
Employees table:

```
+-------------+---------------+------------+
| employee_id | employee_name | manager_id |
+-------------+---------------+------------+
| 1           | Boss          | 1          |
| 3           | Alice         | 3          |
| 2           | Bob           | 1          |
| 4           | Daniel        | 2          |
| 7           | Luis          | 4          |
| 8           | Jhon          | 3          |
| 9           | Angela        | 8          |
| 77          | Robert        | 1          |
+-------------+---------------+------------+
```

Output:

```
+-------------+
| employee_id |
+-------------+
| 2           |
| 77          |
| 4           |
| 7           |
+-------------+
```

Explanation:
The head of the company is the employee with employee_id 1.
The employees with employee_id 2 and 77 report their work directly to the head of the company.
The employee with employee_id 4 reports their work indirectly to the head of the company 4 --> 2 --> 1.
The employee with employee_id 7 reports their work indirectly to the head of the company 7 --> 4 --> 2 --> 1.

The employees with employee_id 3, 8, and 9 do not report their work to the head of the company directly or indirectly.

```
# Write your MySQL query statement below
-- WITH CTE1 AS (SELECT employee_id
--               FROM Employees
--               WHERE manager_id=1 AND employee_id!=1),
-- CTE2 AS (SELECT employee_id
--         FROM Employees
--         WHERE manager_id IN
--                     (SELECT *
--                     FROM CTE1))

-- SELECT employee_id
-- FROM Employees
-- WHERE manager_id IN
--               (SELECT *
--               FROM CTE2)
--     OR employee_id IN
--               (SELECT *
--               FROM CTE2)
--     OR employee_id IN
--               (SELECT *
--               FROM CTE1)

SELECT e1.employee_id
FROM Employees e1
LEFT JOIN Employees e2 ON e1.manager_id = e2.employee_id
LEFT JOIN Employees e3 ON e2.manager_id = e3. employee_id
WHERE e1.employee_id != 1 AND (e1.manager_id=1 OR e2.manager_id=1 OR e3.manager_id=1);
```