```python
from collections import deque

# Read the number of gnomes
N = int(input())

# Read the adjacency matrix
friends = [list(map(int, input().split())) for _ in range(N)]

# Read the number of queries
M = int(input())

for _ in range(M):
    # Read the query parameters
    k, x = map(int, input().split())

    # Create an empty set to store reachable gnomes
    reachable = set()

    # Create an empty queue for BFS
    queue = deque()

    # Enqueue the gnome who created the joke
    queue.append(x)
    reachable.add(x)

    # Perform BFS until k minutes have passed or the queue is empty
    while queue and k > 0:
        gnome = queue.popleft()

        # Enqueue unvisited friends
        for friend in range(N):
            if friends[gnome - 1][friend] == 1 and friend + 1 not in
reachable:
                queue.append(friend + 1)
                reachable.add(friend + 1)

        # Decrement k
        k -= 1

    # Print the size of the reachable set
    print(len(reachable))

    # Print the IDs of the reachable gnomes in increasing order
    if reachable:
        print(' '.join(map(str, sorted(reachable))))
    else:
```

```
        print(-1)
```