You again need to sort an array of N numbers in non-decreasing order. This time, you don't know the values of the array elements, but you are given a helper function that can handle groups of element comparison requests. The helper function has the following signature:

```
void ask(const vector<int>& lhs, const vector<int>& rhs, vector<int>& result)
```

The *lhs*, *rhs* and *result* vectors must have the same length. The vectors *lhs* and *rhs* must contain the indices of the array elements. After calling the function, the *result* vector will contain the results of comparing the corresponding elements of the array, namely:

- If the element at index lhs_i less than element at index rhs_i, then $result_i = -1$
- If the element at index lhs_i greater than element at index rhs_i, then $result_i = 1$
- If the element at index lhs_i equals the element at index rhs_i, then $result_i = 0$

You only need to implement the *solve* function, which takes one parameter N – the size of the array ($1 \leq N \leq 100$).

The function must return a vector containing the indexes of the array elements (elements are indexed from zero), in the desired order.

Your program can call the *ask* function no more than 50 times, otherwise it will get «Wrong answer».