CREATE -TARGET-ARRAY-IN GIVEN ORDER

LEETCODE:ACCORDING TO INDEX INSERT ELEMENTS
Given two arrays of integers nums and index. Your task is to create *target* array under the following rules:

- Initially *target* array is empty.
- From left to right read nums[i] and index[i], insert at index index[i] the value nums[i] in *target* array.
- Repeat the previous step until there are no elements to read in nums and index.

Return the *target* array.

It is guaranteed that the insertion operations will be valid.

Example 1:

```
Input: nums = [0,1,2,3,4], index = [0,1,2,2,1]
Output: [0,4,1,3,2]
Explanation:
nums       index     target
0            0        [0]
1            1        [0,1]
2            2        [0,1,2]
3            2        [0,1,3,2]
4            1        [0,4,1,3,2]
```

Example 2:

```
Input: nums = [1,2,3,4,0], index = [0,1,2,3,0]
Output: [0,1,2,3,4]
Explanation:
nums       index     target
1            0        [1]
2            1        [1,2]
```

```
3            2        [1,2,3]
4            3        [1,2,3,4]
0            0        [0,1,2,3,4]
```

Example 3:

```
Input: nums = [1], index = [0]
Output: [1]
```

Constraints:

- 1 <= nums.length, index.length <= 100
- nums.length == index.length
- 0 <= nums[i] <= 100
- 0 <= index[i] <= i

APPROACH
I WILL TAKE AN EXTRA VECTOR AND USING INBUILT C++ STL FUNCTION "INSERT"
SYNTAX FOR INSERTING AT PARTICULAR INDEX IN C++ IS..
VECTOR_NAME.INSERT(VECTOR_NAME.BEGIN()+(PLACE A NUMBER HERE SUCH THAT FROM BEGINING WHEN ADDED WILL BE THE CORRECT POSITION),VALUE);
HANDLING THE CORNER CASE THAT IS CONSTRAINT GIVEN LENGTH OF BOTH IS SAME
SO I WILL SAY IF INDEX ARRAY HAVING A VALUE GREATER THAN LENGTH OF NUMBER ARRAY
THEN SIMPLY ADD THAT NUMBER ARRAY ELEMENT AT THE END OF TARGET ARRAY.

```
class Solution {
public:
    vector<int> createTargetArray(vector<int>& nums, vector<int>& index) {
        vector<int> ans;
        int n=nums.size();
        for(int i=0;i<n;i++){
            if(index[i]>n){
                ans.push_back(nums[i]);
            }
```

```
            else{

            ans.insert(ans.begin()+index[i],nums[i]);

            }
        }
        return ans;

    }
};
```