The Full Counting Sort

Use the counting sort to order a list of strings associated with integers. If two strings are associated with the same integer, they must be printed in their original order, i.e. your sorting algorithm should be *stable*. There is one other twist: strings in the first half of the array are to be replaced with the character `-` (dash, ascii 45 decimal).

Insertion Sort and the simple version of Quicksort are stable, but the faster in-place version of Quicksort is not since it scrambles around elements while sorting.

Design your counting sort to be stable.

Example
$arr = [[0,'a'],[1,'b'],[0,'c'],[1,'d']]$

The first two strings are replaced with '-'. Since the maximum associated integer is 1, set up a helper array with at least two empty arrays as elements. The following shows the insertions into an array of three empty arrays.

```
i       string  converted       list
0                               [[],[],[]]
1       a       -               [[-],[],[]]
2       b       -               [[-],[-],[]]
3       c                       [[-,c],[-],[]]
4       d                       [[-,c],[-,d],[]]
```

The result is then printed: **- c - d** .

Function Description

Complete the *countSort* function in the editor below. It should construct and print the sorted strings.

countSort has the following parameter(s):

- *string arr[n][2]:* each *arr[i]* is comprised of two strings, *x* and *s*

Returns
- Print the finished array with each element separated by a single space.

Note: The first element of each $arr[i]$, x, must be cast as an integer to perform the sort.

Input Format

The first line contains n, the number of integer/string pairs in the array arr.
Each of the next n contains $x[i]$ and $s[i]$, the integers (as strings) with their associated strings.

Constraints

$1 \leq n \leq 1000000$
n is even
$1 \leq |s| \leq 10$
$0 \leq x < 100, x \in ar$
$s[i]$ consists of characters in the range $\mathbf{ascii[a\text{-}z]}$

Output Format

Print the strings in their correct order, space-separated on one line.

Sample Input

```
20
0 ab
6 cd
0 ef
6 gh
4 ij
0 ab
6 cd
0 ef
6 gh
0 ij
4 that
3 be
0 to
1 be
5 question
1 or
2 not
4 is
2 to
4 the
```

Sample Output

```
- - - - - to be or not to be - that is the question - - - -
```

Explanation

The correct order is shown below. In the array at the bottom, strings from the first half of the original array were replaced with dashes.

```
0 ab
0 ef
0 ab
0 ef
0 ij
0 to
1 be
1 or
2 not
2 to
3 be
4 ij
4 that
4 is
4 the
5 question
6 cd
6 gh
6 cd
6 gh
```

```
sorted = [['-', '-', '-', '-', '-', 'to'], ['be', 'or'], ['not', 'to'], ['be'], ['-', 'that', 'is', 'the'],
['question'], ['-', '-', '-', '-'], [], [], [], []]
```