Question 1 / 1 (Hanoi Moves)

There are **K** pegs. Each peg can hold discs in decreasing order of radius when looked from bottom to top of the peg. There are **N** discs which have radius 1 to **N**; Given the initial configuration of the pegs and the final configuration of the pegs, output the moves required to transform from the initial to final configuration. You are required to do the transformations in minimal number of moves.

- A move consists of picking the topmost disc of any one of the pegs and placing it on top of anyother peg.
- At anypoint of time, the decreasing radius property of all the pegs must be maintained.

Constraints:
1<= **N**<=8
3<= **K**<=5

Input Format:
N K
2nd line contains **N** integers.
Each integer in the second line is in the range 1 to **K** where the **i**-th integer denotes the peg to which disc of radius **i** is present in the initial configuration.
3rd line denotes the final configuration in a format similar to the initial configuration.

Output Format:
The first line contains **M** - The minimal number of moves required to complete the transformation.
The following **M** lines describe a move, by a peg number to pick from and a peg number to place on.
If there are more than one solutions, it's sufficient to output any one of them. You can assume, there is always a solution with less than 7 moves and the initial confirguration will not be same as the final one.

Sample Input #00:

```
2 3
```

```
1 1
2 2
```

Sample Output #00:

```
3
1 3
1 2
3 2
```

Sample Input #01:

```
6 4
4 2 4 3 1 1
1 1 1 1 1 1
```

Sample Output #01:

```
5
3 1
4 3
4 1
2 1
3 1
```

NOTE: You need to write the full code taking all inputs are from **stdin** and outputs to **stdout**
If you are using "Java", the classname is "**Solution**"