Name Chaining

Objective
Give practice with recursion, brute force, and permutations.

Story
The handle generator has not brought in a sizable influx of cash, so you decide to make an advertising campaign to increase the service usage. It turns out that many denizens of Gameland like word play and word games. You will take some common handles generated and order them such that they form a pretty phrase.

A phrase is pretty if for all pairs of consecutive words the first word ends with the same letter that the second word starts with. For example the phrase "re**d** **d**ad**s** **s**ee**k** **k**araoke". The phrase "thei**r** **d**rea**m** **m**anifest**s** **s**om**e** **e**motion" is not pretty, because of the first two words.

The phrase you generate does not need to make sense. The goal will be to use all the words exactly once in a phrase such that the phrase is pretty.

Problem
Given a list of newline-separated handles create an arrangement such that the following word begins with the same letter the prior word ended with.

Input
Input will begin with a line containing 1 integers, ***n*** ($1 \leq n \leq 12$), representing the number of names to process. The following line will ***n*** space separated handles to use in the phrase. Each handle will be composed strictly of at most 20 lowercase Latin letters.

Each handle will have at least one non-space character.

Output
Output should contain a pretty phrase composed of each of the given handles exactly once. It is guaranteed that at least one pretty phrase exists that uses all the given handles. Any pretty phrase will be accepted as long as all handles are used.

Sample Input	**Sample Output**
4 seek red karaoke dads	red dads seek karaoke
2 stressed desserts	desserts stressed

Explanation

Case 1

“red dads seek karaoke” is pretty because “red” ends with ‘d’, which “dads" begins with. “dads” ends with ‘s’ which begins “seek”. Lastly, “seek” ends with ‘k’ which is the same letter that “karaoke” begins with.

Case 2

Either arrangement of “stressed” and “desserts” is considered pretty.

You could output “stressed desserts” or “desserts stressed”.

Hints

Ugly Approach: We could loop over all the input words to figure out which first word we would use. We could then use a loop to figure out which word we want to use second. The second loop would need to be in the first to make sure that you are considering a valid word pairing. For every word position we would need a loop, but the loop would need to be in the loop of the prior word

```
For ()
        For ()
                For ()
                        For ()
                                ...
```

Recurse: We want to run these for loops within each other. We can create a nesting of for loops by calling a function recursively with in the for loop

```
void f(N)
        For()
                f(N-1)
```

Unrolling the above behavior we would see the same nested for loop behavior

Returns: It's a good idea to return a value to represent if a pretty phrase has been found.

Grading Criteria

- Good comments, whitespace, and variable names
    - 15 points
- No extra input output (e.g. input prompts, “Please enter the number of friends:”)
    - 10 points
- Create a recursive method that takes in the number of words left to process
    - 5 points
- Call the recursive method within a loop within a recursive method
    - 10 points
- Return when early to print only one pretty phrase of names.
    - 10 points
- Programs will be tested on 10 cases
    - 5 points each

No points will be awarded to programs that do not compile using “gcc -std=gnu11 -lm”.

Sometime a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use recursion. ***Without this programs will earn at most 50 points!***

Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.

No partial credit will be awarded for an incorrect case.