

Towers of Hanoi

Assignment Description

You must write an **iterative** function that solves the puzzle of **Towers of Hanoi** for a given n number of disks. Your output should be a list of every move needed in order to move all disks from pin A to pin C.

Input-Output specification

Input is a single integer n .

Example

Input	Output
1	['AC']
3	['AC', 'AB', 'CB', 'AC', 'BA', 'BC', 'AC']

Hints

For example, you can use a dictionary to keep track of where the disks are: `tower["A"]` then shows which drives are currently on pin A. We can now move the top disk from A to B with `tower["B"].append(tower["A"].pop())`. Such a move is only allowed if the top checker of A is smaller than the top checker of B, i.e.: `tower["A"][-1] < tower["B"][-1]`.

We are looking for the shortest way to move the disks. Once you have taken the first step, the subsequent steps are fixed; there is only one possible move at a time to proceed.

Notes

This puzzle can be solved with ease recursively, as in order to move the lowest disk, you need to solve the puzzle for $n - 1$ disks. An example of such a recursive solution is:

```
def hanoi_rekursief(n, bron="A", doel="C", hulp="B"):
    if n == 1:
        zetten = [bron + doel]
    else:
        zetten = hanoi_rekursief(n - 1, bron, hulp, doel)
        zetten += [bron + doel]
        zetten += hanoi_rekursief(n - 1, hulp, doel, bron)
    return zetten
```

With this function for $n = 2$ we have the following solution:

```
hanoi_rekursief(2,"A","C","B") = [ 'AB', 'AC', 'BC' ]
```

You can use this solution in order to test if your solution works.