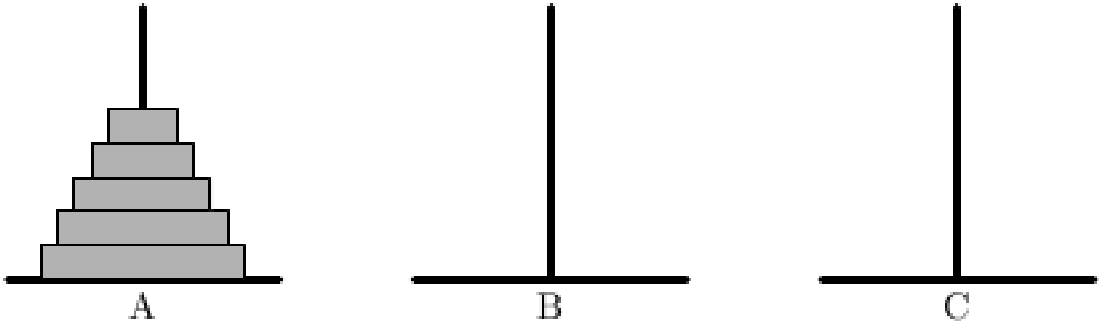
The “Hanoi Towers” puzzle consists of three pegs (that we will name A, B, and C) with n disks of different diameters stacked onto the pegs. Initially all disks are stacked onto peg A with the smallest disk at the top and the largest one at the bottom, so that they form a conical shape on peg A.



A valid move in the puzzle is moving one disk from the top of one (source) peg to the top of the other (destination) peg, with a constraint that a disk can be placed only onto an empty destination peg or onto a disk of a larger diameter. We denote a move with two capital letters - the first letter denotes the source disk, and the second letter denotes the destination disk. For example, `AB` is a move from disk A to disk B.

The puzzle is considered solved when all the disks are stacked onto either peg B (with pegs A and C empty) or onto peg C (with pegs A and B empty). We will solve this puzzle with the following algorithm.

All six potential moves in the game (`AB`, `AC`, `BA`, `BC`, `CA`, and `CB`) are arranged into a list. The order of moves in this list defines our strategy. We always make the first valid move from this list with an additional constraint that we never move the same disk twice in a row.

It can be proven that this algorithm always solves the puzzle. Your problem is to find the number of moves it takes for this algorithm to solve the puzzle using a given strategy.

Input

Input contains several dataset. Each dataset contains two lines. The first line consists of a single integer number n ($1 \leq n \leq 30$) — the number of disks in the puzzle. The second line contains descriptions of six moves separated by spaces — the strategy that is used to solve the puzzle.

Output

For each dataset, write to the output file the number of moves it takes to solve the puzzle. This number will not exceed 10^{18}.

Sample Input

```
3
AB BC CA BA CB AC
2
AB BA CA BC CB AC
```

Sample Output

```
7
5
```