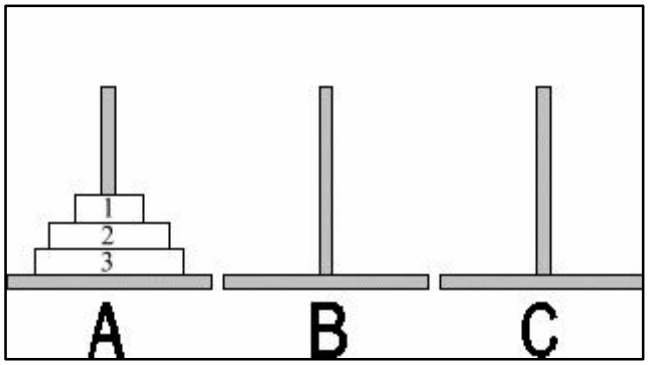
C. Tower of Hanoi (16 Mb, 1 sec)



Let us briefly remind the conditions of this well-known puzzle.

There are 3 rods marked **A**, **B**, **C**. Initially, ***N*** disks of different diameter are stacked in order of size on rod **A,** with the smallest disk at the top. The second and the third rods are empty yet.

The objective is to move all the disks from rod **A** to rod **B**, using rod **C** as auxiliary.

On each move you can take the top disk from any rod and then put it either on an empty rod or on a rod where the top disk is larger than taken.

Virtually all books on programming describe a recursive solution of this puzzle. The following example demonstrates a procedure written in Pascal.

```
Procedure Hanoi (X, Y, Z: char; N: integer);
Begin
 If N>0 then
    Begin
       Hanoi (X, Z, Y, N-1);
       Writeln('Disk ', N, ' from ', X, ' to ', Y);
       Hanoi (Z, Y, X, N-1)
    End
End;
```

Note that when the puzzle is being solved the disks may be moved in the following directions[1]: AB, AC, BA, BC, CA, CB. The example demonstrates the number of movements in each possible direction.

[1] A direction is defined by initial and final rod letters.

Example. N=3

Move No.	Disk	Rod		Direction
		From	To	
1	1	A	B	AB
2	2	A	C	AC
3	1	B	C	BC
4	3	A	B	AB
5	1	C	A	CA
6	2	C	B	CB
7	1	A	B	AB

Direction, S (for N=3)	AB	AC	BA	BC	CA	CB
Number of moves	3	1	0	1	1	1

Write a program to calculate the number of moves for given number of disks N and direction S.

Limitations

1 <= N <= 25

Input

The first line of the input file contains a single integer N, the number of disks, and the second line contains two capital letters S, the direction in question.

Output

The output file must contain a single integer, the number of moves in the given direction S.

Sample Input 1	Sample Output 1
3 **AB**	**3**
Sample Input 2	**Sample Output 2**
3 **CA**	**1**