## Problem 2 – A-nacci

The A-nacci sequence (read "ei nachi"), often called "ei-nachi, aman ot teya zadachi" (read it however you like) is a sequence similar to the Fibonacci sequence – each element is formed by the sum of the previous two, but with a little different rules for the elements.

The elements in the A-nacci sequence are the capital letters from the English alphabet. Each letter has a code, determined by its position in the alphabet – A has the code 1, B has the code 2, …, Z has the code 26. Here are all the elements that can be in an A-nacci sequence, along with their codes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

The first two elements in the sequence can be any two of the letters above. **Every next element** **has a code equal to the sum of the codes of the previous two elements**.

For example, if A and B are the first two elements, the third element will be C (code(A) = 1, code(B) = 2, 1 + 2 = 3, code(C) = 3). Analogically, the fourth element's code will be determined by the sum of the codes of B and C, so the fourth element will be E.

**If the sum of two codes is larger than 26, then that sum is taken by its modulus by 26** (you know this as the % operator in C#). For example, if the sum is 27, then the code will be (27 by modulus 26) = 1, which is the code of A. Another example – if the first two elements are Y and D, then the sum of their codes is 25 + 4 = 29, which is larger than 26, so the code of the next element will be (29 by modulus 26) = 3, which is the code of C.

The **A-nacci figure** consists of **lines of sequential elements from an A-nacci sequence**, printed out similarly to the letter A (but without the dash in the middle). **The first line contains exactly one element – the first element of an A-nacci sequence**. The **second line contains the second and third elements** of the sequence, **concatenated** (that is, not separated by anything). **Each of the next lines contains exactly two elements** – the **next elements of the sequence**, **separated by a certain number of whitespaces**. The number of whitespaces separating the elements on the third line, fourth line and so on, are as follows:

* **The third line has exactly one whitespace** between the two elements
* **The fourth line has exactly two whitespaces** between the two elements
* …
* **The Nth line has exactly N-2 whitespaces** between the two elements

Write a program, which, by given the first two elements (letters) of the A-nacci sequence and the number of lines in the A-nacci form, prints an A-nacci form on the console.

### Input

The input data should be read from the console.

The **first two lines** will contain the values of the **first two elements** of the A-nacci sequence – each element will be a **capital English letter** on a separate line.

On the third line of the input there will be the number **L** – **the number of lines** in the A-nacci form.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output data should be printed on the console.

The output should contain **exactly L lines**. The **first line should contain exactly one capital English letter**. **The second line** (if L>1) **should contain exactly two capital English letters**. The **third line should contain two capital English letters, separated by a single whitespace** (" ") and so on. There **shouldn't be any whitespaces after the second** (i.e. last) **letter on a line**.

### Constraints

* 1 ≤ **L** ≤ 42.
* All elements of the A-nacci sequence are characters, which are capital letters from the English alphabet.
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

### Examples

|  |  |
| --- | --- |
| **Input example** | **Output example** |
| C  B  3 | C  BE  G L |
| A  Q  1 | A |