

Problem F. Fix the Matrix

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

This is an interactive problem. In this problem, you have to respond to the actions of the interactor program.

Firstly, you need to output the matrix H of size 6×6 filled with characters 'A' and 'B'.

In each of the requests the interactor will perform one of the following actions on matrix H to obtain the new matrix T :

1. Change at most one character in each row of matrix H and reorder its rows in an arbitrary manner.
2. Change at most one character in each column of matrix H and reorder its columns in an arbitrary manner.

Operation of character changing means replacing some character 'A' with character 'B' or vice versa.

In each step you are to find out what type of action did the interactor perform. Also, it is necessary to determine for each row or column (depends on the action type) of matrix T its position in the original matrix H .

Note that the **exact order** used by interactor is **required**. For example, if the whole matrix H consists of characters 'A' and interactor changed no characters but reordered some rows, the output with original or any other order, except for the one that is used by the interactor, would be considered incorrect. That is why it does not make sense to use, for example, matrix H with duplicate rows or columns.

You are to write a program that fills the matrix H and responds correctly to all requests from the interactor program.

It is possible that the same matrix can be obtained by performing action of either the first or the second type. In this case you are allowed to output both these variants, one after another. Your response will be considered correct if at least one variant is identical to that performed by the interactor.

Input

After your program has output matrix H , the interactor will start to provide your program with requests. For each request it will perform some type of action on the original matrix H to obtain the result matrix T . Type of action (first or second) and order for rows or columns can be chosen by interactor depending on both matrix H and test case. Each request starts with a line "Request" (quotes for clarity). This line is followed by six lines containing six characters each — the matrix T .

After receiving line "Accepted", your program should be terminated immediately.

It is guaranteed that there will be no more than 100 requests and in each of them the interactor will provide you with a correct matrix T .

Output

Firstly your program should output six lines containing six characters each — the initial matrix H . Each character should be either 'A' or 'B'.

For each of the following requests from the interactor your program should output either two or three lines.

- If there are two possible actions with different action types that lead to the same matrix your program should output three lines. On the first line there should be the only integer 0 (zero)

denoting that there are two possible action types. The second line should describe the possible action of the first type and contain six space-separated integers — i -th integer is the index of i -th row of matrix T in the original matrix H . Similarly, the third line should describe the possible action of the second type and contain six space-separated integers — i -th integer is the index of i -th column of matrix T in the original matrix H .

- If there is the only possible action type your program should output two lines. On the first line there should be the only integer t — the type of action performed by the interactor. The second line should contain six space-separated integers — i -th integer is the index of i -th row or column (depends on the action type) of matrix T in the original matrix H .

All rows are numbered with sequential integers from 1 to 6 from top to bottom, all columns are numbered in the same way from left to right.

Example

standard input	standard output
<i><waiting for output></i>	AAAAAA
<i><waiting for output></i>	BBBBBB
<i><waiting for output></i>	AAABBB
<i><waiting for output></i>	ABABAB
<i><waiting for output></i>	BABABA
<i><waiting for output></i>	BBAAAA
Request	<i><reading input></i>
AAABAA	<i><reading input></i>
BBAAAB	<i><reading input></i>
BBBBBB	<i><reading input></i>
BABABB	<i><reading input></i>
ABABAB	<i><reading input></i>
AAABBA	<i><reading input></i>
<i><waiting for output></i>	1
<i><waiting for output></i>	1 6 2 5 4 3
Request	<i><reading input></i>
AABAAA	<i><reading input></i>
BBBBBB	<i><reading input></i>
AAABBB	<i><reading input></i>
AABBAB	<i><reading input></i>
BBAABA	<i><reading input></i>
BABAAA	<i><reading input></i>
<i><waiting for output></i>	2
<i><waiting for output></i>	1 3 2 4 5 6
Accepted	<i><reading input and terminating></i>