Problem P1 Off the Charts

Your younger brother in second grade has asked you to write a program to chart his scientific experiments over a five day period from Monday to Friday. He would like a horizontal bar chart to reflect a set of non-negative values $1 \le d_i \le 9$ (hey, he is in second grade after all). For example, given "8 2 7 1 3" as input, you must output:

```
M|########
T|##
W|#######
R|#
F|###
```

Your output chart shows the Y-axis as shown above with one character abbreviations for the days of the week, with Monday's value represented on the first line and Friday's value represented on the fifth and final line. Each value d_i is represented by a horizontal bar of hash characters.

Input Format

Your program will read from standard input a single line containing five non-negative integers separated by spaces. You can assume each integer is a value $1 \le d_i \le 9$.

Output Format

Your program will write to standard output five lines of characters. The only valid characters to output are Hash (#), Bar (|) and the capital letters M, T, W, R and F. There must be no spaces in the output.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 8 3 5 1	M|## T|######## W|### R|##### F|#
1 2 9 9 9	M|# T|## W|######### R|######### F|#########
8 2 7 1 3	M|######## T|## W|####### R|# F|###

Problem P2 Falling into Pieces

Given a floating point value *fpv* between 0.11 and 0.999999 with no more than 6 decimal places, find the rational number *a* / *b* (that is, *a* divided by *b*) that most closely approximates this value. Your goal is to find two integers $1 \leq a < 100$ and $1 \leq b < 100$ such that *a* / *b* comes the closest to *fpv* without going over (that is, *a* / *b* is allowed to equal *fpv*). Note that you must compute the reduced form for *a* / *b* when generating your output and *a* and *b* must both be less than 100.

For example, given the value 0.25, you can see that *a*=2 and *b*=8 will exactly equal this value. However 2/8 is not in its most reduced form and you must instead reduce to *a*=1 and *b*=4 to reflect the fraction 1/4.

Some common rational numbers, such as 1/3, have an infinite decimal expansion and do not equal the fixed value of 0.333333 so you should be prepared for some surprises. Given 0.333333 as input, the closest *a* / *b* that approximates this value given the constraints described earlier is 32/97.

Input Format

The input will consist of a single line which will contain a floating point number between 0.11 and 0.999999 with no more than six digits after the decimal point.

Output Format

Your program will write to standard output. Your program should produce a single line of output on a line by itself containing the integers "a b" with a single space separating the two values. Just to be clear, *a* represents the numerator integer, while *b* represents the denominator integer.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
0.2429	17 70
0.345678	19 55
0.25	1 4
0.111111	10 91
0.333333	32 97
0.11	10 91

Problem P3 Let it Snow!

You are given a 5x5 pattern of characters using letters A-Z and the "." character. Based on this pattern you must output a 10x10 pattern reflecting this initial pattern vertically and horizontally. For example, given five rows of characters as shown on the left, you must produce the output of ten-lines as shown on the right. Note how the initial pattern appears as the upper-left corner of the resulting 10x10 output.

Original

```
.....
...A.
.....
.B...
.....
```

Generated Output

```
..........
...A..A...
..........
.B......B.
..........
..........
.B......B.
..........
...A..A...
..........
```

Input Format

The input will consist of five lines, each of which will contain five characters each on a line by itself.

Output Format

Your program will write to standard output. Your program should produce ten lines of output, each of which contains ten characters.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output		Sample Input	Sample Output
ABCDE ABCDE ABCDE ABCDE ABCDE	ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA ABCDEEDCBA		ABBA. A..A. A..AE ACCA. ..D..	ABBA..ABBA A..A..A..A A..AEEA..A ACCA..ACCA ..D....D.. ..D....D.. ACCA..ACCA A..AEEA..A A..A..A..A ABBA..ABBA
A.... .A... ..A.. ...A. ....A	A........A .A......A. ..A....A.. ...A..A... ....AA.... ....AA.... ...A..A... ..A....A.. .A......A. A........A		0...0 ....X.. ..... 0...0	0...00...0 ....X....X.. .... 0...00...0 0...00...0 ....X....X.. .... 0...00...0

Problem P4 Deep Roots

Some strings can be constructed by repeating a base string multiple times. For example, the string ABABABAB can be formed by the repeated concatenation of AB four times; thus it can be considered to be AB^4. Other strings defy having a repeated substring. For example, ABABB is not composed by the repeated concatenation of any specific string so it is considered atomic.

Given a string of characters, *s*, you must compute its base string, *b*, and the largest integer $n \geq 1$ such that $b^n = s$. Note that you must find the largest *n* value if there are multiple candidates. For example, the string ABABABAB shown above can also be considered to be $ABAB^2$ but you most compute the highest root of 4 using the base string AB. For an atomic string, *a*, its base string is equal to itself and the root $n = 1$.

Input Format

Your program will read a single string from standard input on a line by itself. You can be guaranteed that this string will have between 1 and 32 capital letters.

Output Format

Your program will write to standard output. The first line must contain the base string on a line by itself. The second line must contain the integer root on a line by itself.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
ABABABABABAB	AB 6
ABCBABCB	ABCB 2
ABCD	ABCD 1
AAAAA	A 5

Problem P5 Babe

Your task is to convert a sequence of English words into Pig Latin. There are numerous variations of Pig Latin so you must follow the following rules:

- If a word starts with a Vowel (A, E, I, O or U) then simply append "WAY" to the word. Thus ALL becomes ALLWAY and UNTIL becomes UNTILWAY

- If a word begins with any other letter (i.e., a consonant) then identify the *prefix* of the word composed entirely of consonants and extract it and move to the end of the word, followed by "AY". Thus BALL becomes ALLBAY and STRAIGHT becomes AIGHTSTRAY.

You can assume that every word in the sentence has at least an (A, E, I, O, or U). For simplicity you can assume that the letters "Q" and "Y" do not appear in any input.

Input Format

Your program will read from standard input. The only line of input will contain a sentence of words all with capital letters (except the letters Q and Y) and a single space separating each word from its neighbor. The input will contain no more than 64 characters.

Output Format

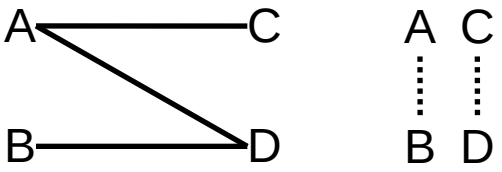
Your program will write to standard output a single line containing the Pig Latin translation, with a single space separating each word from its neighbor. Make sure there are no trailing space characters in your output.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
THIS IS OUR BEST	ISTHAY ISWAY OURWAY ESTBAY
I HAVE A SECRET	IWAY AVEHAY AWAY ECRETSAY
NO BALL WINS	ONAY ALLBAY INSWAY
ONE TWO THREE FOUR FIVE	ONEWAY OTWAY EETHRAY OURFAY IVEFAY
FUN IN THE SUN	UNFAY INWAY ETHAY UNSAY
THESE PRAWNS ARE ENORMOUS	ESETHAY AWNSPRAY AREWAY ENORMOUSWAY

Problem P6 My Cousin Vinny

You are given a simple undirected graph G of vertices and edges. This graph represents those vertices that are directly connected to other vertices. No more than one edge can exist between any two vertices. Two vertices *u* and *v* in G are *cousins* if they are not directly connected to each other but there is a third vertex *x* such that edges *u–x* and *v–x* exist in G. Your task is to count the number of cousin pairs in a graph G.



For example, given the 4-vertex, 3-edge graph G on the far left, there are two cousin pairs: (A, B) and (C, D).

A graph may have no cousin pairs.

Each vertex is labeled by a capital letter in a consecutive range starting with A, B, and so on. Thus a 5-vertex graph has vertices labeled A, B, C, D, and E.

Input Format

The input will consist of a sequence of lines. The first line will contain two integers "n e" representing the number of vertices, *n*, and the number of edges, *e*, in a graph G. Each of the remaining *e* lines contains a 3 character string of the form "u-v" where *u* is a capital letter representing a vertex in the graph that is directly connected to another vertex representing by the capital letter *v*. You can assume that *u* is not equal to *v*, that the input contains no duplicate edges, and that *u* and *v* refer to valid vertices in G. All graphs will have between 3 and 9 nodes and between 2 and 36 edges.

Output Format

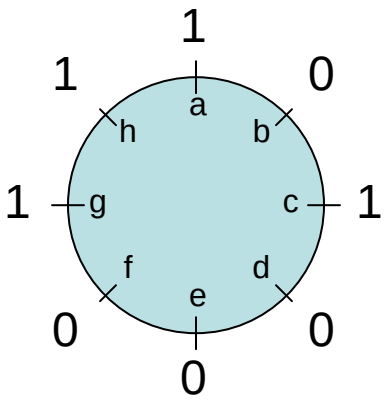
Your program will write to standard output. Your program should produce a single line containing a non-negative number by itself indicating the number of cousin pairs.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
3 2 A-B B-C	1
3 3 A-B B-C A-C	0
5 7 A-B A-C A-D B-C B-E C-E D-E	3

Problem P7 Binary City

Given an integer $n > 0$ there are 2^n possible unique sequences of 0s and 1s you can form of length n. For example, for n=3, you can form the set of string sequences T = {000, 001, 010, 011, 100, 101, 110, and 111}.



A binary *De Bruijn* sequence of order n is a cyclic sequence of 0s and 1s for which every possible sequence of length n appears as a sequence of consecutive characters exactly once. On the left, shown clockwise around a circle, the string "10100011" is a De Bruijn sequence of order 3. You can choose any three consecutive digits in clockwise order and you will find a different string sequence of size 3. If you select the digits labeled 'abc' the sequence is "101" and no other 3 consecutive digits have this value. Selecting digits labeled 'hab' results in the sequence "110" which is also unique on this circle.

Although the existence of such *De Bruijn* sequences is not obvious, it is well known that they exist for all orders n>0. Your task is to read in an integer n and a string of 2^n binary characters (0s and 1s) and determine whether the string represents a binary *DeBruijn* sequence.

Input Format

Your program will read from standard input. The first line will contain a positive integer n on a line by itself. The second line will contain a string s composed of 2^n characters. You can be guaranteed that s only contains "0" and "1" characters. You can assume that $2 \le n \le 6$.

Output Format

Your program will write to standard output a single word ("YES" or "NO") on a line by itself. Make sure the output uses capital letters with no extra spaces or characters.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 1001	YES
3 10100011	YES
3 10100110	NO
5 00000111110111001101011000101001	YES

Problem P8 The Tipping Point

You are given a common die sitting on a table in a specific orientation. It sits on the table in front of you with a 1 showing on its top face, a 2 showing on its front face, and a 3 showing on its right face. Because the die is ordinary you should know that the total of the numbers on opposite sides of the die add up to 7. Thus the die has a 4 on its left face, a 5 on its back face, and a 6 hidden on its bottom face resting on the table.

The die can tip over to reveal a new number based upon the direction of tipping. You can tip the die AWAY, TOWARD, LEFT or RIGHT. The table below shows the result of each tip given the initial placement described earlier.

AWAY	TOWARD	LEFT	RIGHT
2 on top, 6 in front and 3 on right	5 on top, 1 in front and 3 on right	3 on top, 2 in front and 6 on right	4 on top, 2 in front and 1 on right

Given the initial die configuration and a sequence of tip instructions, you must determine the number that remains on the top face of the die after all tips have been completed.

Input Format

Your program will read from standard input an integer on a line by itself. This integer represents the number of tip instructions that are to follow, one per line. Each successive line of input contains a single word in upper case reflecting the type of move, either AWAY, TOWARD, LEFT or RIGHT.

Output Format

Your program will write to standard output. Your program should produce a single line of output. The line will contain an integer value 1, 2, 3, 4, 5, or 6 on the line by itself.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 AWAY AWAY	6
6 AWAY LEFT LEFT TOWARD RIGHT TOWARD	2
4 AWAY AWAY AWAY AWAY	1
3 LEFT TOWARD RIGHT	1

Problem P9 Shuffle off to Buffalo

Given a string containing an even number of characters, you are asked to perform a number of perfect *faro* shuffles on the string. In a perfect faro shuffle, the string is divided into a left and right half of the original string. Then the characters are alternatively merged with each other to reform a string of the same original length.

There are two types of faro shuffles. An **out** faro shuffle leaves the original first character at the front and the original final character at the end. An **in** shuffle moves the original first character to be second and the original final character to be second from the last.

out faro shuffle **in** faro shuffle

For the string "ABCD1234", the **in** and **out** shuffle are shown above. You are given an initial string, *s*, and a sequence of **in** and **out** faro shuffle commands. Your task is to output *s* after all shuffles have completed.

Input Format

Your program will read from standard input a single line that will contain the base string, which will consist of an even number (between 4 and 20) of alphanumeric characters on a line by itself. The second line of input will contain a positive integer, *n*, on a line by itself representing the number of faro shuffles to perform. The next *n* lines of input will each contain a single word, "in" or "out" (in lower case), representing the type of faro shuffle to execute. The order of the *n* lines reflects the order of the shuffles to be executed.

Output Format

Your program will write to standard output. Your program should produce a single line of output on a line by itself which contains a permutation of characters drawn from the original base string, representing the state of the string after all shuffles have executed.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output		Sample Input	Sample Output
ABCD1234 2 in in	31CA42DB		ABCDEFGH12345678 4 in in out in	65872143FEHGBADC
ABCD1234 3 out out out	ABCD1234		ABCDEFGH12345678 3 out in out	EGAC5713FHBD6824

Problem P10 I, Robot

You are asked to simulate the behavior of a robot with three robotic arms known as LEFT, MIDDLE and RIGHT. The robot will perform a series of commands and then it must be able answer a few simple questions.

Commands		**Questions**	**Answer**
pick up XXX block with YYY hand		is your YYY hand empty?	YES or NO
drop block in YYY hand on table		is there anything directly on top of the XXX block?	YES or NO
place block in YYY hand on XXX block		what is the color of the block directly on top of the XXX block?	XXX
		what is the color of the block in your YYY hand?	XXX
		is the XXX block directly on the table?	YES or NO

XXX will be a valid color:
RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, and VIOLET

YYY will be a valid hand:
LEFT, MIDDLE, and RIGHT

The robot can only pick up blocks that have no block on top of them, and the robot will only be given commands that can be performed. The robot "knows" how to pick up blocks found on a flat table surface. There are seven blocks each with a different color. All are the same size. A block can either be dropped on the table (there is plenty of space on the table) or placed on top of another block (you can assume that towers of any size can stand).

Input Format

Your program will read from standard input a series of lines. The first line will contain the words "`BEGIN COMMANDS`" on a line by itself. Each successive line will contain a command on a line by itself according to the syntax of the above table until a line of input reads "`BEGIN QUESTIONS`". The remaining lines from input will each contain a single question on a line by itself according to the syntax of the above table until a line of input reads "`END QUESTIONS`".

Output Format

Your program will write to standard output. Your program should produce a series of strings, one per line, that represent the answers to the questions from the input. As shown in the table above, there are five possible question types but there are only two types of answers: either "YES" or "NO" should be output, or one of the seven colors.

Problem P10 I, Robot

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
BEGIN COMMANDS pick up RED block with MIDDLE hand pick up BLUE block with LEFT hand pick up GREEN block with RIGHT hand place block in LEFT hand on VIOLET block place block in MIDDLE hand on BLUE block pick up INDIGO block with MIDDLE hand place block in MIDDLE hand on ORANGE block BEGIN QUESTIONS is your MIDDLE hand empty? is your RIGHT hand empty? is the YELLOW block directly on the table? what is the color of the block directly on top of the BLUE block? is there anything directly on top of the VIOLET block? is there anything directly on top of the BLUE block? is there anything directly on top of the ORANGE block? END QUESTIONS	YES NO YES RED YES YES YES
BEGIN COMMANDS pick up RED block with LEFT hand pick up BLUE block with RIGHT hand place block in RIGHT hand on YELLOW block place block in LEFT hand on BLUE block pick up GREEN block with RIGHT hand BEGIN QUESTIONS is your MIDDLE hand empty? what is the color of the block directly on top of the YELLOW block? what is the color of the block in your RIGHT hand? is the VIOLET block directly on the table? END QUESTIONS	YES BLUE GREEN YES
BEGIN COMMANDS pick up RED block with LEFT hand place block in LEFT hand on BLUE block pick up GREEN block with RIGHT hand place block in RIGHT hand on RED block BEGIN QUESTIONS is the GREEN block directly on the table? what is the color of the block directly on top of the BLUE block? END QUESTIONS	NO RED