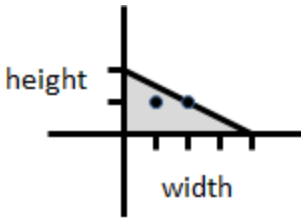
P1 :: Right Triangulation



You are given integer values describing the height and width of a right triangle oriented in the lower left corner of quadrant I of the Cartesian plane as shown here. Your task is to determine the number of points (i, j) contained within this triangle where i and j are integers greater than zero. A point (i, j) that lies exactly on the edge of the hypotenuse is considered to be contained within the triangle. In this example, there are two such points contained within the triangle whose height is 2 and width is 4.

Input Format

Your program will read from standard input. The first line will contain a single positive integer by itself representing the height. The second line will contain a single positive integer by itself representing the width. Both numbers will be less than or equal to 50.

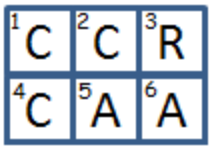
Output Format

Your program must write to standard output a single integer on a line by itself representing the total number of points (i, j) contained within the right triangle where $i > 0$ and $j > 0$.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 4	2
1 10	0
5 5	10
10 10	45
3 7	6

P2 :: Car Finder



Given an MxN array of letters containing just the letters ‘C’, ‘A’ and ‘R’, your task is to count the number of different times the word CAR appears when starting from a ‘C’ letter and traveling to an adjoining letter (in one of potentially 8 neighboring directions, as with the game BOGGLE®) and finally ending up at an ‘R’ letter. In the 2x3 example on the right you can spell CAR four different ways. Using the small numbers in each cell, CAR appears as:

(1,5,3), (4,5,3), (2,5,3) and (2,6,3).

M and N are guaranteed to be positive integers less than or equal to 10.

Input Format

Your program will read from standard input. The first line will contain a single positive integer by itself representing the number of rows, M. The second line will contain a single positive integer by itself representing the number of columns, N. The next successive M lines of input will contain strings of N characters representing the array. These lines will only contain the capital letters ‘C’, ‘A’ and ‘R’.

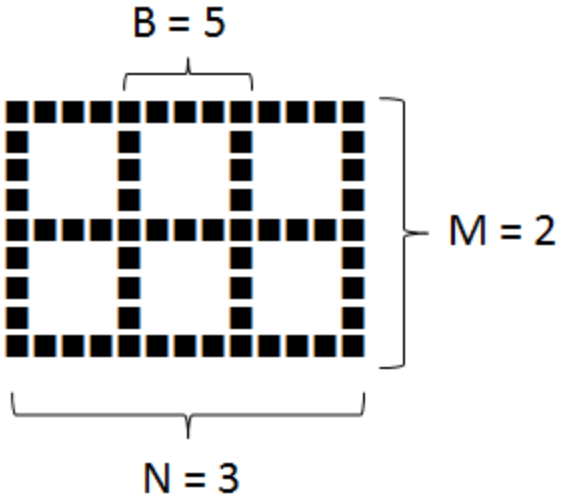
Output Format

Your program must write to standard output a single integer on a line by itself representing the total number of different ways to spell CAR in the array.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 2 CC AR	2
2 3 ACR ACR	0
3 3 CCR CAA AAA	4
2 5 CACAC ARARA	8

P3 :: Counting Pixels



Consider printing an MxN rectangular grid of square boxes where each box has height and width of B pixels. Each black dot (■) on the printed grid at right represents a single pixel and M=2, N=3, and B=5. You can assume that $0 < M, N \leq 10$ and $1 < B \leq 10$.

Your task is to compute the total number of black dots to be used in printing the grid. In this sample, the total is 63.

Input Format

Your program will read from standard input. The first line will contain a single positive integer by itself representing the size of each square box, B. The second line will contain a single positive integer by itself representing the number of rows, M, in the grid. The third line will contain a single positive integer by itself representing the number of columns, N, in the grid.

Output Format

Your program must write to standard output a single integer on a line by itself representing the total number of pixels used to print the grid.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
5 1 2	27
4 3 3	64
4 4 3	82
2 1 1	4

P4 :: Time Clock

You need to compute the total time a worker has worked on a single calendar day given two timestamps of the form “HH:MM”. The numbers MM can be in the range of “00” to “59” while HH is in the range “01” through “12”. This is an archaic timestamp clock that doesn’t even record AM or PM values. The system has worked well in the past because no-one has ever worked longer than an 8 hour shift.

You can be assured that the two times are different and represent an employee checking in to work (on a calendar day) and checking out from work later on the exact same calendar day. You are to output the total time as “HH:MM”. If the accumulated work time is greater than 8 hours (in other words, 481 minutes or longer), then you are ordered to output “08:00” because this is the maximum time that the employee would be paid.

For example, given timestamps of “09:13” and “04:42” the total reported working time is reported as “07:29”.

Input Format

Your program will read from standard input. The first line will contain a single timestamp representing the starting time containing five characters of the form “HH:MM” where HH is in the range “01” to “12” while MM is in the range “00” through “59”. The second line will contain a single timestamp representing the ending time in the same “HH:MM” format.

Output Format

Your program must write to standard output the accumulated time as “HH:MM” on a single line by itself where HH represents the number of hours in the range “00” to “12” and MM represents the number of minutes in the range “00” to “59”.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
09:13 04:42	07:29
09:12 10:03	00:51
03:10 11:15	08:00
07:13 01:01	05:48

P5 :: Math Series

The following function $f(x)$ is defined for all integers greater than 1.

$$f(x) = \begin{cases} 6x + 1, & if\ x\ is\ prime \\ \sum_{p_i \mid x} f(p_i)\ where \prod_{i=1}^{n} p_i = x, & otherwise \end{cases}$$

When *x* is prime, $f(x)$ simply computes to be 6*x* + 1; otherwise, find the unique prime factorization of x, $\{p_1, p_2, \ldots, p_n\}$, such that $x = p_1 \times p_2 \times \ldots \times p_n$. Then compute $f(x) = \sum_{i=1}^{n} f(p_i)$.

For example, $f(13)$ = 6*13+1 = 79 because 13 is a prime. To compute $f(120)$, observe that the factorization of 120 = $2 \times 2 \times 2 \times 3 \times 5$. Thus

$$f(120) = f(2) + f(2) + f(2) + f(3) + f(5)$$
$$f(120) = 13 + 13 + 13 + 19 + 31$$
$$f(120) = 89$$

You can assume that 1 < *x* < 10000.

Input Format

Your program will read a line from standard input that contains a single positive integer by itself representing *x*.

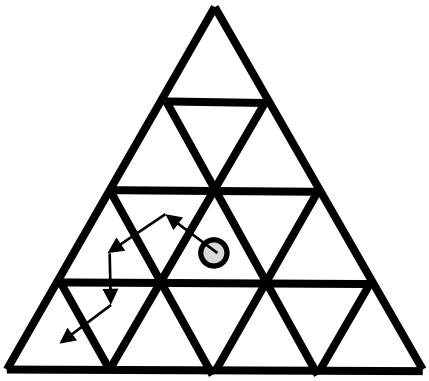
Output Format

Your program must write to standard output a single integer on a line by itself representing $f(x)$.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
17	103
120	89
8	39
72	77
117	117

P6 :: The $100,000 Pyramid



Imagine a pyramid formed by 16 triangles as shown on right where you can slide a token from a starting triangle (identified by a gray circle) by visiting neighbors along one of its three edges. In the example on right, the path goes upper left (UL), then lower left (LL) then down (D) then lower left (LL). The final destination is the lower corner triangle.

There are six possible moves: From a triangle, you can visit upper left (UL), upper right (UR) or down (D). From an inverted (or upside down) triangle, you can visit lower left (LL), lower right (LR) or up (U). Given a sequence of such moves from the initial triangle marked by the gray circle, you must determine if the final destination is the initial triangle from which the sequence started. You can assume that each move is a valid move and that you never "fall off" the board. Each sequence will have no more than 10 moves.

Input Format

Your program will read from standard input. The first line will contain a positive integer on a line by itself representing the number of moves in the sequence, N. The second line will contain N move representations, separated by whitespace. These moves will be described in upper case letters and will be one of {UL, UR, D, LL, LR, U}.

Output Format

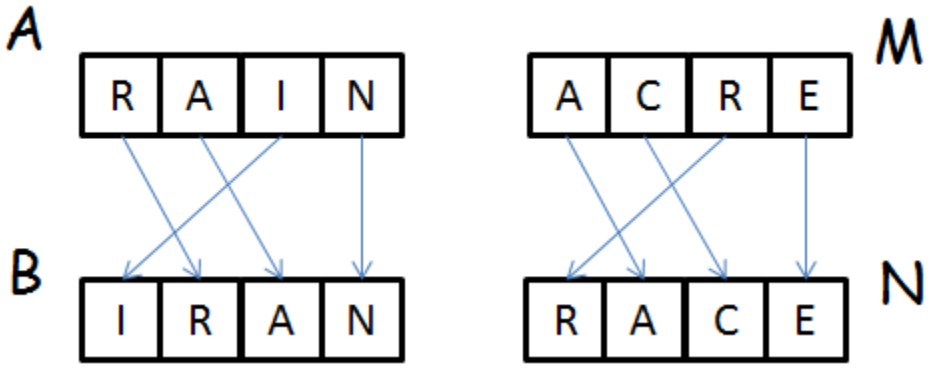
You must output a single string on a line by itself. If the sequence of moves brings you back to the initial square, output the string INITIAL otherwise output the string DIFFERENT.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
4 UL LR D LL	DIFFERENT
6 UR LR D LL UL U	INITIAL
10 D U D LR UR LL UL LL UR U	INITIAL
3 UR U UL	DIFFERENT
9 UR LR D LL UL LL UL U UR	DIFFERENT

P7 :: Analogous Issues

You are given an analogy of the form A::B, where B is an anagram of word A. You must compute the same analogy for a similar sized word M by rearranging the letters of M to create a new word N in the same way as described by the analogy.



For example, given the analogy RAIN::IRAN above, the similar analogy for the word ACRE is RACE. Note how the shuffling of letters from A to B is replicated to create N from M. The word A contains no duplicate letters. The same is true of B and M. All strings (A, B, M and N) will have between 2 and 10 characters each.

Input Format

Your program will read from standard input. The first line will contain the analogy in the form of two strings A and B of the same length separated by two colons ("::"). The second line will contain a string M of the same length as A and B.

Output Format

Your program must write to standard output a single string on a line by itself representing the analogy for the input word M. The length of this output string must be the same as the length of M.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
SMART::TRAMS GRATE	ETARG
TRAINS::STRAIN FATHER	RFATHE
STRANGE::AGENTRS SUBTLED	TEDLUBS
LEASH::SHALE FRUIT	ITUFR
RAIN::IRAN ACRE	RACE

P8 :: DNA Binding

Biologists commonly model nucleotide strands (such as RNA and DNA) using sequences of nucleotide bases. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). DNA is formed by two individual strands whose bases align to form pairs where A pairs with T and G pairs with C.

You are given a target nucleotide strand sequences and you must determine the *binding site* where this strand can bind in its entirety with a contiguous subsequence of a source strand. The following shows the proper alignment of target strand CTGCATC against source strand GTGACGTAGC because of the C/G pairs and A/T pairs.

```
GTGACGTAGC
..CTGCATC.
```

You can be assured that there will be no more than one possible alignment for the two input strands given in this problem. No input strand will contain more than 20 bases and the target strand will always be shorter than the source strand.

Input Format

Your program will read from standard input. The first line will contain the target nucleotide sequence. The second line will contain the source nucleotide sequence. The only characters that appear in the input are characters A, T, G and C.

Output Format

If there is a binding then your program must produce two lines of output. The first line must contain the source strand on a line by itself. The second line must contain the target strand with the appropriate number of leading periods (".") to indent strand one so it aligns perfectly with the binding site. Make sure you also include trailing (".") characters as needed so both output lines have the exact same number of characters.

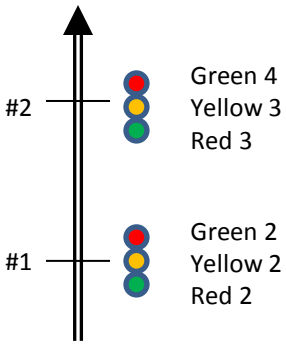
If there is no binding site, then your program must output NO BINDING on a single line of output by itself.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
CTGCATC GATCGTGACGTAGCTGATCG	GATCGTGACGTAGCTGATCG .....CTGCATC.....
AAAAAA CGTGCGTGCT	NO BINDING
ATTATTC TAATAAGCA	TAATAAGCA ATTATTC..
GACT ATGACTGA	ATGACTGA ....GACT

P9 :: One, Two Three… Red Light!

A long street in New York City has a number of traffic lights at successive intersections, all in a row. Each light starts Green for a fixed number of seconds, then turns Yellow for a fixed number of seconds, then turns Red for a fixed number of seconds. This cycle repeats for each light using its own set of timings. When all lights are turned on, they start at the beginning of their cycle with a Green light. Given N lights, with their respective timings, compute the earliest time (in seconds) when all lights show Red.



In the example above, light #1 turns Red at the 5th second, but when this happens, light #2 is Yellow. At the 8th second, light #2 is Red but light #1 has cycled through and is Green again. You can verify that in the 18th second, both lights are Red.

You can assume the total maximum cycle (adding Green, Yellow and Red times) is 99 seconds. You can assume $0 < N < 5$ and each Green, Yellow and Red time period is at least 1 second long.

Input Format

Your program will read from standard input. The first line will contain a positive integer on a line by itself representing the number of lights, N. The next N lines represent the N lights. Each line of input will contain three positive integers separated by spaces – G, Y and R – representing the number of seconds that light is Green, then Yellow, then Red.

Output Format

You must output a single positive integer on a line by itself representing the earliest time (in seconds) when all N lights show red.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
2 2 2 2 4 3 3	18
3 10 5 2 30 4 10 20 10 30	169
1 13 2 1	16
4 40 5 20 12 5 13 17 2 11 9 9 78	50

P10 :: Anagramania

Given two lines containing only upper-case letters and white space, determine whether the second line is a complete *anagram* of the first line without missing any letters or adding extra letters. An anagram of a phrase is a different phrase that uses the exact same letters but in a different order. Note that whitespace is not considered part of the anagram.

For example, given the phrase CLINT EASTWOOD one valid anagram is OLD WEST ACTION. Each line of input will have fewer than 50 characters (including white space).

Input Format

Your program will read from standard input. The first line will contain a string of upper-case letters and whitespace representing the original phrase. The second line will contain a string of upper-case letters and whitespace representing the proposed anagram.

Output Format

If the second line is a proper anagram of the first, output "ANAGRAM" on a line by itself, otherwise output "NO ANAGRAM"

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
GEORGE HEINEMAN OH GAME ENGINEER	ANAGRAM
TRAIN TRACKS RATS ART NICK	ANAGRAM
THIS IS A TEST TESTS ARE OUT	NO ANAGRAM
A PROGRAMMING CONTEST A SCREAMING TONG TROMP	ANAGRAM