به یاد استاد فقید

دکتر سید مهدی فخرایی

(۱۳۹۳—۱۳۳۹)

استاد تمام دانشکده مهندسی برق و کامپیوتر دانشگاه تهران

و از پیشگامان حوزه ابر رایانه و میکروالکترونیک ایران

مضارب ٦

می خواهیم در یک بازه از اعداد، اعدادی را که مضرب ۶ هستند بیابیم. برنامه ای بنویسید که دو عدد صحیح را به عنوان ابتدا و انتهای بازه بگیرد و کلیه اعدادی که در این بازه که مضرب ۶ هستند در خروجی چاپ نماید.

ورودی برنامه:

دو عدد صحیح (می توانند منفی نیز باشد)

خروجی برنامه:

همه اعداد مضرب ۶ در آن بازه

نمونه ورودی	نمونه خروجی
7 21	**12** **18**
-9 17	**-6** **0** **6** **12**

.B

دایره های تو در تو

در یک صفحه مختصات دو بعدی دو دایره داریم. این دایره ها ممکن است همدیگر را قطع کرده باشند، با هم مماس باشند، یکی درون دیگری باشد و یا اینکه دو دایره جدا از هم باشند. در این برنامه هدف ما این است که با داشتن مختصات مراکز دو دایره و شعاع هر یک، مشخص کنیم که آیا یکی از دایره ها به طور کامل درون دیگری است یا خیر.

ورودی برنامه:

در ابتدا سه عدد اعشاری x1 و y1 و r1 که مختصات مرکز دایره اول و نیز شعاع دایره اول از کاربر گرفته می شود. سپس سه عدد اعشاری x2 و y2 و r2 که مختصات و شعاع دایره دوم هستند از کاربر گرفته می شود.

شعاع یک عدد غیر منفی می باشد.

خروجی برنامه:

در خروجی برنامه یکی از کلمات Yes (برای حالتی که یکی به طور کامل درون دیگری باشد) یا No (برای حالات دیگر) چاپ شود.

نکته: مماس بودن دو دایره نیز No تلقی می شود.

نمونه ورودی	نمونه خروجی
0,0,2 0,4,3	**No**
5,6,1 **1,1,2**	**No**
1,1,7 **2,2,1**	**Yes**

.C

حبیب و رباتش

حبیب اسکندری که عضو انجمن علمی علوم کامپیوتر است که به خاطر علاقه زیاد به علم هوش مصنوعی یک ربات ساخته است. حرکت این ربات را می توان با فرمان هایی که بصورت اس ام اس برای آن فرستاده می شود کنترل کرد. حبیت برای تست رباتش، آن را درون یک جدول ۱۰ در ۱۰ (با ۱۰۰ خانه) قرار داده که خانه های آن از صفر تا ۹۹ شماره گذاری شده اند. سپس یک اس ام اس حاوی یک رشته از کاراکترها برای ربات می فرستد و در نهایت شماره خانه ای که ربات به آن رسیده را ارزیابی می کند تا ببیند آیا فرمان هایش را به درستی اجرا می کند یا خیر. این رشته ترکیبی است از تعدادی کاراکترهای L، R، U، D و S که پشت سر هم قرار می گیرند. ربات دریافت این رشته و تحلیل کاراکترهای آن، به صورت زیر حرکت می کند:

L: حرکت به طرف خانه سمت چپ. R: حرکت به طرف خانه سمت راست. U: حرکت به طرف خانه بالایی. D: حرکت به طرف خانه پایینی. S: ایستادن و حرکت نکردن.

اگر ربات در سمت چپی ترین خانه باشد و کاراکتر L را ببیند امکان حرکت ندارد و همانجا باقی می ماند. برای کاراکترهای R و U و D هم شرایط مشابهی وجود دارد. در واقع در صورت رسیدن به لبه های جدول امکان حرکت در آن امتداد وجود نخواهد داشت.

در نهایت هدف بدست آوردن موقعیت نهایی ربات پس از اجرای دستورات داده شده می باشد.

90	91	92	93	94	95	96	97	98	99
80	81	82	83	84	85	86	87	88	89
70	71	72	73	74	75	76	77	78	79
60	61	62	63	64	65	66	67	68	69
50	51	52	53	54	55	56	57	58	59
40	41	42	43	44	45	46	47	48	49
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

ورودی برنامه:

در ابتدا یک عدد صحیح (بین صفر و ۹۹) از ورودی گرفته می شود که شماره خانه ای است که ربات در ابتدا در آن قرار دارد. سپس یک رشته حاوی کاراکترهای بزرگ L ، R، U، D و S از ورودی گرفته می شود. رشته ورودی دارای حداقل ۱ کاراکتر و حداکثر ۵۰ کاراکتر خواهد بود.

خروجی برنامه:

خروجی برنامه یک عدد است که شماره خانه ای را نشان می دهد که ربات پس از اجرای فرمان به آن خواهد رسید.

نمونه ورودی	نمونه خروجی
15 **LLURSUSU;**	**44**
58 **UUSRRRRLDSD**	**58**
2 **USDDLLLLUSSD**	**0**

.D

Digi-Eater

Sasuke is the owner of one of the most curious monsters in the world: The Digi-eater. A Digi-eater is a monster that can be fed with numbers every once in a while. When a Digi-eater eats a number, it evolves multiple times. This particular monster is unique because during evolution, the Digi-eater adds the digits of the number in his belly and eats the sum, evolving again and again until the number in his belly has only one digit. This final number is engraved on his digital belly.

For example, Sasuke fed his Digi-eater with the number 987 and saw that it evolved 2 times. Adding all the digits, 9 + 8 + 7, the Digi-eater changed the number to 24. It then ate this number, changed the number to 2 + 4 = 6, leaving his digi-eater 2 levels up in evolution and with a funny number 6 printed on his belly.

Help Sasuke figure out the number that will be printed on his Digi-eater's belly before he feeds it again.

Input:

The input is an integer Number N ($1 \leq N \leq 1{,}000{,}000$).

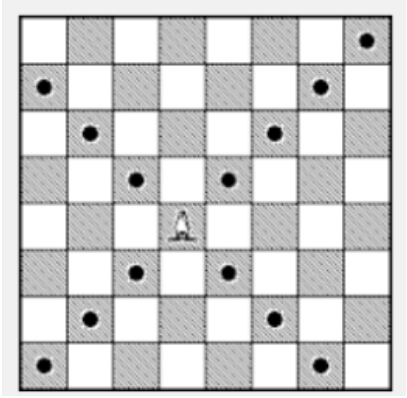
Output:

Print the resulting number.

Sample Input	Sample Output
10	**1**
33	**6**
987	**6**
10000	**1**
123123	**3**

.E

Chess



In Konoha Academy, Shikamaru has set up a new challenge for all the ninja students. He sets up a chess board with two bishops at different locations. The task is to find all the squares where they could intercept each other in a single move. (A bishop moves along two diagonals in any direction until it reaches the edge of the board, as shown in the figure to the left.

Knowing Shikamaru's prowess at mind games, the ninja students have come to you for help!

Input

The input is a line with four integers x1, y1, x2, y2 representing the location of the first and second bishop. (Location 0, 0 is at the lower left corner of the board.) (Constraints: $0 \leq x1, y1, x2, y2 < 500$.) The Initial location of the bishops will always be different, and they will not be located on the same diagonal. Assume that the bishops will always intersect at at least one square.

Output

The output is a line containing either two or four integers, which represent the coordinates of the locations where the bishops intersect (at either one or two locations). The coordinates with the smaller x value should appear first. If the x-values are same, then display the coordinates having the smaller y-value first. You should only output those intersections having positive values for both x and y.

Sample Input and Output

input	output
0 0 2 0	1 1
2 2 2 4	1 3 3 3

Konoha Training

Saitama is a transfer student and new member of Konoha Academy. He is learning some hand seals and just mastered a fire technique named Punch. This technique can be described as a linear beam of force projected from his body towards a specific target. While training, his professor will select a target walking x steps horizontally from Saitama, and y steps vertically.

Last week Saitama's professor leveled up his training by creating an invisible wall-shield on the field from point (X1,Y1) to (X2,Y2). Only the professor knows its location, but since the field is a huge terrain (and there could be casualties), he wants to know beforehand if Saitama will hit the target. We will assume that striking the endpoint of a wall also qualifies as striking the wall. If the target is located on the shield, the Punch will count as a hit.

Input

The input consists of a line with six integers, x1, y1, x2, y2, x, y. Therefore, (x1, y1) and (x2, y2) are the endpoints of the wall-shield, and (x, y) is the target's position.

Saitama is always at position (0, 0) and $0 \leq |x1|, |y1|, |x2|, |y2|, |x|, |y| \leq 10^6$

Output

Print "YES", if Saitama can hit the target, or "NO" in case the shield will stop the Punch.

Sample Input and Output

Input	Output
2 3 3 2 3 3	NO
-1 1 1 1 1 -1	YES
1 1 2 2 3 3 2 2 3 3 1 1	NO YES
0 1 1 1 0 2	NO

Word Counting

You've been tasked with building a major component of a word processing program. Given a line and a word, you are to determine the number of occurrences of that word in the line. Note that the word can be contained within another word in the same line, and these should count as two separate occurrences. Two words are considered equal if they contain the same characters in the same order, ignoring uppercase/lowercase differences.

Input

Each test case will be 2 lines, the first line will contain the word that we want to find the occurrences of and the line following that will be the sentence where we want to count these occurrences. Words will be no longer than 30 characters and sentences will not be longer than 600 characters.

Output

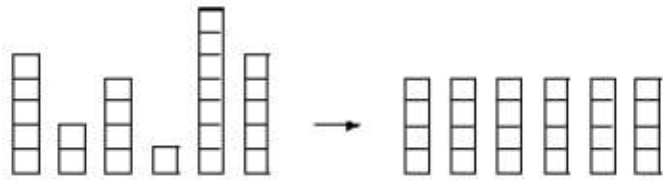
The output is the number of occurrences of that word in the sentence.Each output should

Sample Input and Output

Input	Output
`the` `the problem is too easy`	`1`
`hArd` `This problem isn't that hard. I wonder if they get any Harder...`	`2`

Box of Bricks

Little Bob likes playing with his box of bricks. He puts the bricks one upon another and builds stacks of different height. "Look, I've built a wall!", he tells his older sister Alice. "Nah, you should make all stacks the same height. Then you would have a real wall.", she retorts. After a little consideration, Bob sees that she is right. So he sets out to rearrange the bricks, one by one, such that all stacks are the same height afterwards. But since Bob is lazy he wants to do this with the minimum number of bricks moved. Can you help?



Input

The input consists of several data sets. Each set begins with a line containing the number n of stacks Bob has built. The next line contains n numbers, the heights hi of the n stacks. You may assume 1 <= n <= 50 and 1 <= hi <= 100. The total number of bricks will be divisible by the number of stacks. Thus, it is always possible to rearrange the bricks such that all stacks have the same height.

Output

Print the minimum number of bricks that have to be moved in order to make all the stacks the same height.

Sample Input and Output

Input	Output
6 5 2 4 1 7 5	5
4 5 5 5 5	0

.I

Permutations

A permutation of a string is the set of all possible ways to combine its characters. E.g., the permutation of "abc" is {"abc", "acb", "bac", "bca", "cab", "cba"}. The size of this set is the factorial of the initial string size.

Given a string S (with up to 10 characters, all lowercase letters) and a integer N ($0 < N < 10!$) find the (N+1)th smallest element of the permutation of S (consider the lexicographic order; the permutation of 'abc' above, for example, is represented in lexicographic order form left to right).

For example,

if S = "abc" and N=0, then the result would be "abc"

if S = "abc" and N=5, then the result would be "cba"

if S = "abc" and N=3, then the result would be "bca"

if S = "cba" and N=3, then the result would be "bca"

Notice that the string may not be initially sorted (check the last two examples).

Input

Each test case consists of two lines: one containing the string S and the next line containing the number N.

Output

The output is a line containing the $(N+1)^{th}$ smallest element of the permutation of S.

Sample Input and Output

Input	Output
abc 3	bca
abcde 119	edcba
cba 3	bca

An Industrial Spy

Industrial spying is very common for modern research labs. I am such an industrial spy, don't tell anybody! My recent job was to steal the latest inventions from a famous math research lab. It was hard to obtain some of their results but I got their waste out of a document shredder.

I have already reconstructed that their research topic is fast factorization. But the remaining paper snippets only have single digits on it and I cannot imagine what they are for. Could it be that those digits form prime numbers? Please help me to find out how many prime numbers can be formed using the given digits.

Input

Each test case consists of a single line. This line contains the digits (at least one, at most seven) that are on the paper snippets.

Output

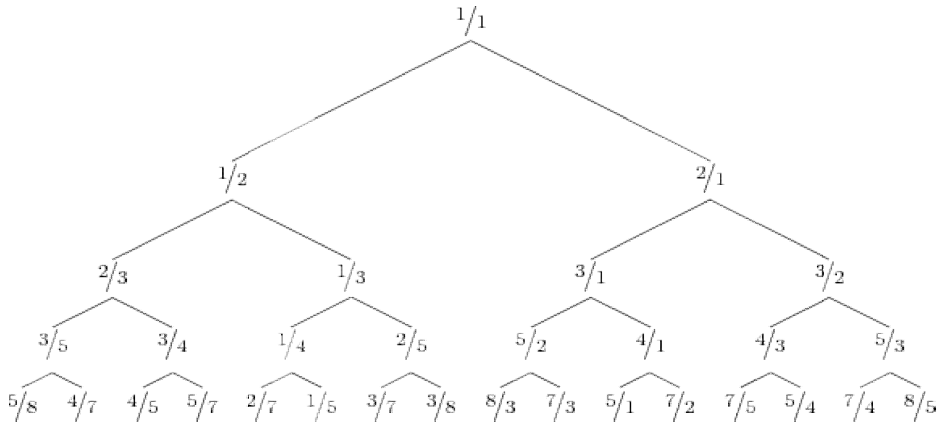
Print one line containing the number of different primes that can be reconstructed by shuffling the digits. You may ignore digits while reconstructing the primes (e.g., if you get the digits 7 and 1, you can reconstruct three primes 7, 17, and 71). Reconstructed numbers that (regarded as strings) differ just by leading zeros, are considered identical (see the fourth case of the sample input).

Sample Input and Output

Input	Output
17	3
1276543	1336
9999999	0
011	2

Bird tree

The Bird tree is an infinite binary tree, whose first 5 levels look as follows:



It can be defined as follows:

$$bird = \begin{array}{c} 1/1 \\ \hline 1/(bird+1) \quad (1/bird)+1 \end{array}$$

This is a co-recursive definition in which both occurrences of bird refer to the full (infinite) tree.The expression bird + 1 means that 1 is added to every fraction in the tree, and 1/bird means that every fraction in the tree is inverted (so a/b becomes b/a). Surprisingly, the tree contains every positive rational number exactly once, so every reduced fraction is at a unique place in the tree. Hence, we can also describe a rational number by giving directions (L for left sub tree, R for right sub tree) in the Bird tree. For example, 2/5 is represented by LRR. Given a reduced fraction, return a string consisting of L's and R's: The directions to locate this fraction from the top of the tree.

Input

For each test case, One line with two integers a and b (1 <=a; b <= 109), separated by a '/'. These represent the numerator and denominator of a reduced fraction. The integers a and b are not both equal to 1, and they satisfy gcd(a; b) = 1.

For every test case the length of the string with directions will be at most 10 000.

Output

For each test case:

One line with the string representation of the location of this fraction in the Bird tree

Sample Input and Output

Input	Output
1/2	L
2/5	LRR
7/3	RLLR