Digit Sum 3

The program's input is a three-digit number (do not check that fact, just assume it to be true). Output the sum of its digits.

Sample Input	Sample Output
151	7
787	22
202	4

Area of a right triangle

The input of the program consists of two integers: the legs of a right triangle. Output the triangle's area.

Sample Input	Sample Output
4 5	10.0
4 4	8.0
5 5	12.5

Arithmetic Progression

The program input consists of 3 integer numbers: a1, a2 and n. a1 and a2 are the first two members of arithmetic progression. Output the value of the n-th member.

2 4 3	6
3 3 1500	3
9 4 2	4

Century from year

Given a year, return the century it is in. The first century spans from the year 1 up to and including the year 100, the second - from the year 101 up to and including the year 200, etc.

Sample Input	Sample Output
1905	20
1700	17
1988	20
2000	20
2001	21
200	2
374	4
45	1

Two men

Two men start to shoot several cans of Coca-Cola they have put on a log. The first man began shooting the cans in order, starting with the

leftmost, the second man from the rightmost. At some point, they simultaneously shot the same last can. And at this point they stop.

You are given the number of cans the first man has shot, and the number of cans the second man has shot. Output the number of cans the first man missed because of the second man, and the number of cans the second man missed because of the first man.

Sample Input	Sample Output
4 7	6 3

Knight's Possible Moves

You are given the coordinates of a cell on a standard chess board: py and px. It is guaranteed that the coordinates are correct, i.e. are integers from the interval [1,8]. Output all cells that the knight can move in a single move(each coordinate pair on separate line). It is guaranteed that for a given input cell all 8 moves exist. The output cells order does not matter.

Sample Input	Sample Output
6 4	5 2 7 2 4 3 4 5 5 6 7 6 8 3 8 5

Digit Sum 2

The program's input is a two-digit number (do not check that fact, just assume it to be true). Output the sum of its digits.

Sample Input	Sample Output
15	6
78	15
20	2

Trinomial

The program input consists of 4 integer numbers: a, b, c, x_o .Output the value of a trinomial ax^2+bx+c in the point x_0.

Sample Input	Sample Output
1 1 1 -2	3
-1 2 1 3	-2
5 6 7 0	7

Round Number

The program's input is a float positive number. Round up the given number to the closest integer and output.
Important: Do not use ready function round().

Sample Input	Sample Output
2.7	3

11.2	11
0.5	1
0.1	0

Swap

Complete the missing part of code.

```
a = input()
b = input()

print("Initial value of variable a:", a)
print("Initial value of variable b:", b)

# Your code starts here

...

# Your code ends here

print("Swapped value of variable a:", a)
print("Swapped value of variable b:", b)
```

Sample Input	Sample Output
hello world	Initial value of variable a: hello Initial value of variable b: world Swapped value of variable a: world Swapped value of variable b: hello

Balls

The box contains multi-colored balls. All balls are the same shape, size and weight. Balls can be one of N different colors. There are many balls of each color in the box.

The program input consists of one positive integer: N - the number of possible colors. Output the minimum number of balls that can be pulled out of the box without looking, so that among them there are guaranteed to be two balls of the same color. It is guaranteed that N >= 1.

Sample Input	Sample Output
3	4

Arithmetic Progression 2

The input of the program consists of five integer numbers: n, a_n, m, a_m and k. a_n is the n-th member of an arithmetic progression, a_m is the m-th. You may assume that all n, m and k are positive, n < m and the progression consists only of whole numbers. Output the k-th member of the progression.

Hint: Use the equations $a_n=a_1 + (n-1)d$ and $a_m=a_1 + (m-1)d$ to get the values a_1 and d, after which use the same formula $a_k=a_1 + (k-1)d$ to obtain the needed value. Use new variables to store any intermediate results.

Sample Input	Sample Output
4 4 7 7 9	9.0
2 9 4 17 1	5.0

1 10 5 30 4	25.0

Digit Product

Input a natural n and output the product of its *nonzero* digits.

Input	Output
45947	5040
301	3
170009	63

Largest Power of 3

Given a natural number N, output the largest integer power of 3 which does not exceed N.

9	9
100	81
500	243
1	1

Triangle

You are given three whole numbers a, b, and c. If there doesn't exist a non-degenerate triangle with such sides, output "No Triangle". If the triangle exists and all its angles are acute, output "Acute Triangle". If the

triangle exists and has a right angle, output “Right Triangle”. If the triangle exists and has an obtuse angle, output “Obtuse Triangle.”

Sample Input	Sample Output
4 3 5	Right Triangle
2 1 1	No Triangle
2 7 8	Obtuse Triangle
4 4 4	Acute Triangle

The Root of the Number

The root of a number is
- the sum of its digits if that sum is less than 10
- the root of the sum of its digits otherwise.

Let’s consider 78996. The sum of its digits is 39. Since it’s not less than 10, we have to find the root of 39. The sum of its digits is 12. Still not less than 10, so repeating again, we add the digits again and get 3, which is finally less than 10. So the root of 78996 is 3. Given a natural number N, find its root, outputting intermediary results in the process, as shown in the samples.

78996	78996 39 12 3
16	16 7

55	55 10 1

Salaries

Given the salaries of three employees working at a department, find the amount of money by which the salary of the highest-paid employee differs from the salary of the lowest-paid employee. The input consists of three positive integers - the salaries of the employees. Output a single number, the difference between the top and the bottom salaries

Sample Input	Sample Output
100 500 1000	900
500 100 1000	900
36 11 20	25
20 20 20	0

Boring Numbers

A natural number is said to be boring if all its digits are the same. Determine if the given number is boring.

Input	Output
777777	Boring
6	Boring

666	Boring
6655	Interesting

Largest Number

You are given a natural number. If it is possible to rearrange/shuffle its digits and get a larger number than the one you started with, output “Yes”. Otherwise, output “No”. For example, given 3112 you can rearrange the digits and get 3211, which is larger than 3112, hence the answer is Yes. In contrast, no matter how you rearrange the digits of 987, you will not be able to get a larger number, hence the answer for 987 should be No.

Sample Input	Sample Output
2	No
678	Yes
88889	Yes
99888740	No
9414	Yes

Line Segment Intersection

You are given four real numbers- a1, b1, a2, b2 - The endpoints of two line segments on a line. Find the length of their intersection. Note that the order of the endpoints of a segment is irrelevant, i.e. the segments [1;2] and [2;1] are considered the same.

Sample Input	Sample Output
1 4 9 7	0

1 2.5 3 2	0.5
10 0 0.1 0.2	0.1

Number Of Divisors

Input a positive x. Find the number of divisors of that x.
(Hint: you can check if y is a divisor for x by checking x%y == 0)

Input	Output
1	1
3	2
10	4

Quadratic Equation

Input three real numbers a, b, c and solve the equation $ax^2+bx+c=0$.

Output an information about whether the equation is

1. Quadratic equation
2. Non-quadratic equation

For quadratic equations output the value of it's discriminant
And in any case output the number of solutions, and the solutions themselves. Follow the examples below.
D = b * b - 4 * a * c
X1,2 = (-b +- sqrt(D)) / 2 * a

Sample Input	**Sample Output**

1.5 -2 4	Quadratic equation Discriminant: -20 No Solutions
0 0 7	Non-quadratic equation No Solutions
0 -2 1	Non-quadratic equation One solution: 0.5
1 0 -1	Quadratic equation Discriminant: 4 Two solutions: 1 -1
0 0 0	Non-quadratic equation Infinite solutions

The Goldbach Conjecture

The Goldbach's Conjecture states that any positive even integer greater than two can be written as a sum of two primes. Given a positive even n not exceeding 10000, output two prime numbers such that their sum equals n. Write a function that checks whether its argument is prime.

4	2 2
992	73 919
16	3 13
20	3 17

Palindrome numbers

Input 2 positive numbers a and b. It's guaranteed that a <= b. Print all the numbers from interval [a, b], that are the same from both sides. Those numbers are called palindromes. Maybe you should reverse the number, e. g. reverse 105 and get 501.

8 25	8 9 11 22
100 120	101 111
100 150	101 111 121 131 141
13000 13500	13031 13131 13231 13331 13431

Suffix Sums

You're given a sequence of real numbers A. Print a sequence B so that, i-th element of the sequence B is equal to the sum of all elements in sequence A starting from i-th.

Вход	Выход
1.5 2.5 3	7.0 5.5 3.0
1 2 3 4 5	15.0 14.0 12.0 9.0 5.0

Cyclic shift

A cyclic shift of a sequence to the right is a sequence that is built by shifting the last element of an initial sequence to the beginning. Given numbers N and k and a sequence of whole numbers of length N, output the result of consecutive shifting to the right k times.

5 2 1 2 3 4 5	4 5 1 2 3

6 1 7 1 7 2 6 1	1 7 1 7 2 6
5 5 9 100 6 0 1	9 100 6 0 1
3 1000 9 1 2	2 9 1

Index Sum

You are given two sequences. The first is a sequence A of N real numbers (N > 0), the other is a sequence of indices IND of size M (all the elements of the second sequence are nonnegative integers strictly less than N). Output the sum of elements of A with indices from IND. For example, if A = [1,2,3,4,5] and IND = [0 3 3 2], you must calculate the sum A[0]+A[3]+A[3]+A[2] = 1 + 4 + 4 + 3 = 12.
The first line of input contains a positive integer N, followed by N real numbers. The third line contains a nonnegative integer M followed by M nonnegative integers strictly less than N.

Sample Input	Sample Output
5 -5 2.5 0 1 3 4 0 0 4 4	-4.0
2 1.5 2.5 3 1 1 1	7.5

The most divisor-rich number

The input consists of two positive integers a and b, such that a <= b. Among the integers in the interval [a, b], find the one that has the most number of divisors. Write a function that finds the number of divisors for a given number.

30 40	36
10 100	60
1000 2000	1680

Lucky Numbers

Let’s call a number lucky if the sum of its digits on even positions is equal to the sum of its digits on odd positions. The positions are numbered starting from 1 from the right end of the number - the last digit having position 1, the penultimate having position 2, etc. Let’s consider 15224. The sum of digits on odd positions is 4 + 2 + 1 = 7. The sum of digits on even positions is: 2 + 5 = 7. Therefore, 15224 is a lucky number. Given a natural number, determine if it’s lucky.

15224	Yes
53143277	Yes
10	No
8	No
121	Yes
212	No

Monotonicity

The input is a whole positive number N (N>1) and a sequence of N numbers. Output Ascending, if the sequence is strictly ascending, Descending, if it’s strictly descending and Neither, if it’s neither ascending, nor descending.

Input	Output
5 1 2 5 7 9	Ascending

6 1 1 2 5 7 9	Neither
4 3 2 1 -10	Descending
4 3 2 2 1	Neither
5 1 2 1 3 4	Neither
2 1 2	Ascending
2 1 1	Neither

Tree

Input the base width of the tree(number of '*' in the bottom) and draw tree with symbols *. You're guaranteed that the number is odd. You have to decide how many spaces and '*' to print in first line, and print them, then how many spaces and '*' to print in the second line and so on.

5	<pre> * *** *****</pre>
9	<pre> * *** ***** ******* *********</pre>

Bayan

Problem statement: http://acm.timus.ru/problem.aspx?space=1&num=1563&locale=en

Ice Cream Parlor

5 points only for O(t * n) solution. (t-is the number of queries, read about the variable n in tasks description)
Problem statement: https://www.hackerrank.com/challenges/ctci-ice-cream-parlor/problem

Beautiful binary string

Problem statement: https://www.hackerrank.com/challenges/beautiful-binary-string/problem

String power

Suppose you have a string s and an integer k. If k is nonnegative, then s^k is a string resulting in k-fold concatenation of s to itself. If k is negative, then s^k is a string x such that x^{-k} is equal to s. Obviously, if k is negative s^k is not always defined.

Given a string s and an integer k, output s^k if it is defined and the word “undefined” otherwise

ab 1	ab
abc 0	
abcd 3	abcdabcdabcd
xyzxyz -2	xyz
xyzxyz -3	undefined
xyzxyz -1	xyzxyz

Super Reduced String

statement-https://www.hackerrank.com/challenges/reduced-string/problem

Strong password

statement-https://www.hackerrank.com/challenges/strong-password/problem

Two strings

statement-https://www.hackerrank.com/challenges/two-strings/problem

Jewels and Stones

statement-https://leetcode.com/problems/jewels-and-stones/

Number of Good Pairs - https://leetcode.com/problems/number-of-good-pairs/
Unique Number of Occurrences - https://leetcode.com/problems/unique-number-of-occurrences/

Distribute Candies - https://leetcode.com/problems/distribute-candies/

Making Anagrams - https://www.hackerrank.com/challenges/ctci-making-anagrams/problem

Find Words That Can Be Formed by Characters - https://leetcode.com/problems/find-words-that-can-be-formed-by-characters/