# Exercise: Basic Syntax

This document defines the exercises for the ["C++ Fundamentals" course @ Software University](https://softuni.bg/trainings/4263/cpp-fundamentals-november-2023).

Please submit your solutions (source code) of all below-described problems in [Judge](https://judge.softuni.org/Contests/2939/CPlusPlus-Basic-Syntax).

# Order Two Numbers

Write a program that:

* Read **two integers** from the console
* Print the **two numbers in increasing order**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 | 1 2 |
| 1 -1 | -1 1 |
| 4242 1313 | 1313 4242 |

# Product Sign

Write a program that shows the sign **(+ or -)** of the product of three real numbers without calculating it.

* Read **3 real numbers** from the console (on a single line, separated by spaces)
* Print the sign of their product (if the product is 0, print '+')

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 0 | + |
| 1 -1 1 | - |
| -411531.13 123123 -8673.24 | + |

# Quadratic Equation

Write a program that enters the coefficients **a**, **b,** and **c** of a quadratic equation **a \* x2 + b \* x + c = 0** and calculates and prints its real solutions. Note that quadratic equations may have **0**, **1,** or **2** real solutions.

You can check your program against this: <https://www.mathsisfun.com/quadratic-equation-solver.html>

The numbers **a**, **b**, and **c** will be entered on a single line from the console, separated by spaces.

* If the quadratic equation has no real roots (e.g. if the Discriminant is less than 0), print "**no roots**".
* If it has one real root print it.
* If it has two roots, print them on a single line, separated by a single space.

On the first place, print the root calculated by formula **(- b + sqrt(D)) / 2**

Оn the second place, print the root calculated by the formula **(- b - sqrt(D)) / 2**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Explanation** |
| 2 5 -3 | -3 0.5 | Equation: 2x2 + 5x - 3 = 0 |
| 10 1 3 | no roots | Equation: 10x2 + x + 3 = 0 |
| 0.5 5 12.5 | -5 | Equation: 0.5x2 + 5x + 12.5 = 0 |

# Numbers 1 to N

Write a program that:

* Read the **integer number N** from the console
* Print **all numbers from 1 to N (inclusively)** to the console on a single line

**Note:** **The number N will always be larger than or equal to 1.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | 1 |
| 10 | 1 2 3 4 5 6 7 8 9 10 |

# Min and Max

Write a program that:

* Reads an **integer number N**
* Then reads a **line of N integers**
* Print the **minimum** and **maximum** of those integers, separated by single space

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  -1 5 | -1 5 |
| 7  5 3 44 21 69 2 10 | 2 69 |

# Greatest Common Divisor

### Write a program that calculates the greatest common divisor (GCD) of given two numbers

* Read **two integer numbers** on a single line from the console, separated by a single space
* **Find** their GCD (Greatest Common Divisor)
* **Print** their GCD (Greatest Common Divisor)

### Hint: you can use the Euclidean algorithm.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Explanation** |
| 25 10 | 5 | 5 is the largest number that divides both 25 and 10 (without a remainder) |
| 50 50 | 50 | Both numbers are 50, so GCD is 50 |
| 7 13 | 1 | 7 and 13 are prime numbers, meaning they only divide by 1 and themselves, so their GCD is 1 |

## Print and Sum

Write a program that:

* Read an **integer number (start number)** on the first line from the console
* Read an **integer number (end number)** on the second line from the console
* Print numbers from given start number to given end number
* Print their sum in the following format: **"Sum: {sum}"**

**Note: All the numbers will be integers.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  10 | 5 6 7 8 9 10  Sum: 45 |
| 0  26 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26  Sum: 351 |
| 50  60 | 50 51 52 53 54 55 56 57 58 59 60  Sum: 605 |

## Strong Number

Write a program that:

* Reads an **integer number N**
* Check whether a given number is **strong**
  + Number is strong if the sum of the Factorial of each digit is equal to the number.

**For example:** 145 is a strong number, because **1! + 4! + 5! = 145**

* Print "**yes**" if the number is **strong**
* Print "**no**" if the number is **NOT strong**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | yes |
| 3451 | no |
| 40585 | yes |