# Homework: Console Input / Output

This document defines homework assignments from the [“C# Basics“ Course @ Software University](http://softuni.bg/courses/csharp-basics/). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code only) of all below described problems.

## Sum of 3 Integers

Write a program that reads **3 integer numbers** from the console and prints their **sum**. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **sum** |
| 3 | 4 | 11 | 19 |
| -2 | 0 | 3 | 1 |
| 5.5 | 4.5 | 20.1 | 30.1 |

## Print Company Information

A company has **name, address, phone number, fax number, web site and manager**. The manager has **first name, last name, age and a phone number**. Write a program that reads the information about a company and its manager and prints it back on the console.

## Circle Perimeter and Area

Write a program that reads the radius **r** of a circle and prints its perimeter and area formatted with 2 digits after the decimal point. Examples:

|  |  |  |
| --- | --- | --- |
| **r** | **perimeter** | **area** |
| 2 | 12.57 | 12.57 |
| 3.5 | 21.99 | 38.48 |

## Number Comparer

Write a program that gets **two numbers** from the console and prints the greater of them. Try to implement this without **if** statements. Examples:

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **greater** |
| 5 | 6 | 6 |
| 10 | 5 | 10 |
| 0 | 0 | 0 |
| -5 | -2 | -2 |
| 1.5 | 1.6 | 1.6 |

## Formatting Numbers

Write a program that reads 3 numbers: an integer a (0 ≤ a ≤ 500), a floating-point b and a floating-point c and **prints them in 4 virtual columns** on the console. Each column should have a width of 10 characters. The number a should be printed in **hexadecimal, left aligned**; then the number a should be printed in binary form, padded with zeroes, then the number b should be **printed with 2 digits after the decimal point, right aligned**; the number c should be **printed with 3 digits after the decimal point, left aligned**. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **result** |
| 254 | 11.6 | 0.5 | |FE |0011111110| 11.60|0.500 | |
| 499 | -0.5559 | 10000 | |1F3 |0111110011| -0.56|10000 | |
| 0 | 3 | -0.1234 | |0 |0000000000| 3|-0.123 | |

## Quadratic Equation

Write a program that reads the coefficients **a**, **b** and **c** of a quadratic equation **ax2 + bx + c = 0** and solves it (prints its real roots). Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **roots** |
| 2 | 5 | -3 | x1=-3; x2=0.5 |
| -1 | 3 | 0 | x1=3; x2=0 |
| -0.5 | 4 | -8 | x1=x2=4 |
| 5 | 2 | 8 | no real roots |

## Sum of 5 Numbers

Write a program that **enters 5 numbers** (given in a single line, separated by a space), **calculates and prints their sum**. Examples:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **numbers** | **sum** |  | **numbers** | **sum** |  | **numbers** | **sum** |
| 1 2 3 4 5 | 15 | 10 10 10 10 10 | 50 | 1.5 3.14 8.2 -1 0 | 11.84 |

## Numbers from 1 to n

Write a program that reads an integer number **n** from the console and prints all the numbers in the interval [**1**..**n**], each on a single line. Note that you may need to use a for-loop. Examples:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **numbers** | **sum** |  | **numbers** | **sum** |  | **numbers** | **sum** |
| 3 | 1  2  3 | 5 | 1  2  3  4  5 | 1 | 1 |

## Sum of n Numbers

Write a program that enters a number **n** and after that enters more **n** numbers and calculates and prints their sum. Note that you may need to use a for-loop. Examples:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **numbers** | **sum** |  | **numbers** | **sum** |  | **numbers** | **sum** |
| 3  20  60  10 | 90 | 5  2  -1  -0.5  4  2 | 6.5 | 1  1 | 1 |

## Fibonacci Numbers

Write a program that reads a number n and prints on the console the first n members of the [**Fibonacci sequence**](http://en.wikipedia.org/wiki/Fibonacci_number) (at a single line, separated by spaces) : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, …. Note that you may need to learn how to use loops. Examples:

|  |  |
| --- | --- |
| **n** | **comments** |
| 1 | 0 |
| 3 | 0 1 1 |
| 10 | 0 1 1 2 3 5 8 13 21 34 |

## \* Numbers in Interval Dividable by Given Number

Write a program that reads two positive integer numbers and prints how many numbers **p** exist between them such that the reminder of the division by **5** is **0**. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **start** | **end** | **p** | **comments** |
| 17 | 25 | 2 | 20, 25 |
| 5 | 30 | 6 | 5, 10, 15, 20, 25, 30 |
| 3 | 33 | 6 | 5, 10, 15, 20, 25, 30 |
| 3 | 4 | 0 | - |
| 99 | 120 | 5 | 100, 105, 110, 115, 120 |
| 107 | 196 | 18 | 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, 170, 175, 180, 185, 190, 195 |

## \*\* Falling Rocks

Implement the "**Falling Rocks**" game in the text console. A small dwarf stays at the bottom of the screen and can move left and right (by the arrows keys). A number of rocks of different sizes and forms constantly fall down and you need to avoid a crash.

Rocks are the symbols **^**, **@**, **\***, **&**, **+**, **%**, **$**, **#**, **!**, **.**, **;**, **-** distributed with appropriate density. The dwarf is **(O)**. Ensure a constant game speed by **Thread.Sleep(150)**.

Implement collision detection and scoring system.

