**Homework: Loops**

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.

* **1.Numbers from 1 to N**

Write a program that enters from the console a positive integer **n** and **prints all the numbers from 1 to n**, on a single line, separated by a space. Examples:

|  |  |
| --- | --- |
| **n** | **output** |
| 3 | 1 2 3 |
| 5 | 1 2 3 4 5 |

* **2.Numbers Not Divisible by 3 and 7**

Write a program that enters from the console a positive integer **n** and prints all the **numbers from 1 to n not divisible by 3 and 7**, on a single line, separated by a space. Examples:

|  |  |
| --- | --- |
| **n** | **output** |
| 3 | 1 2 |
| 10 | 1 2 4 5 8 10 |

* **3.Min, Max, Sum and Average of N Numbers**

Write a program that reads from the console a sequence of **n** integer numbers and returns the **minimal**, the **maximal** number, the sum and the average of all numbers (displayed with 2 digits after the decimal point). The **input** starts by the number **n** (alone in a line) followed by **n lines**, each holding an integer number. The **output** is like in the examples below. Examples:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **input** | **output** |  | **input** | **output** |
| 3  2  5  1 | min = 1  max = 5  sum = 8  avg = 2.67 | 2  -1  4 | min = -1  max = 4  sum = 3  avg = 1.50 |

* **4.Print a Deck of 52 Cards**

Write a program that generates and prints **all possible cards from a** [**standard deck of 52 cards**](http://en.wikipedia.org/wiki/Standard_52-card_deck) (without the jokers). The cards should be printed using the classical notation (like 5♠, A♥, 9♣ and K♦). The card faces should start from 2 to A. Print each card face in its four possible suits: clubs, diamonds, hearts and spades. Use 2 nested **for**-loops and a **switch**-**case** statement.

|  |
| --- |
| **output** |
| 2♣ 2♦ 2♥ 2♠  3♣ 3♦ 3♥ 3♠  …  K♣ K♦ K♥ K♠  A♣ A♦ A♥ A♠ |

* **5.Calculate 1 + 1!/X + 2!/X2 + … + N!/XN**

Write a program that, for a given two integer numbers **n** and **x**, calculates the sum S = 1 + 1!/x + 2!/x2 + … + n!/xn. Use only one loop. Print the result with 5 digits after the decimal point.

|  |  |  |
| --- | --- | --- |
| **n** | **x** | **S** |
| 3 | 2 | 2.75000 |
| 4 | 3 | 2.07407 |
| 5 | -4 | 0.75781 |

* **6.Calculate N! / K!**

Write a program that calculates **n! / k!** for given **n** and **k** (1 < **k** < **n** < 100). Use only one loop. Examples:

|  |  |  |
| --- | --- | --- |
| **n** | **k** | **n! / k!** |
| 5 | 2 | 60 |
| 6 | 5 | 6 |
| 8 | 3 | 6720 |

* **7.Calculate N! / (K! \* (N-K)!)**

In combinatorics, the number of ways to choose **k** different members out of a group of **n** different elements (also known as the number of [**combinatio HYPERLINK "http://en.wikipedia.org/wiki/Combination"ns**](http://en.wikipedia.org/wiki/Combination)) is calculated by the following formula:



For example, there are 2598960 ways to withdraw 5 cards out of a standard deck of 52 cards. Your task is to write a program that calculates **n! / (k! \* (n-k)!)** for given **n** and **k** (1 < **k** < **n** < 100). Try to use only two loops. Examples:

|  |  |  |
| --- | --- | --- |
| **n** | **k** | **n! / (k! \* (n-k)!)** |
| 3 | 2 | 3 |
| 4 | 2 | 6 |
| 10 | 6 | 210 |
| 52 | 5 | 2598960 |

* **8.Catalan Numbers**

In combinatorics, the [Catalan numbers](http://en.wikipedia.org/wiki/Catalan_number) are calculated by the following formula:



Write a program to calculate the **nth Catalan number** by given **n** (1 < n < 100). Examples:

|  |  |
| --- | --- |
| **n** | **Catalan(n)** |
| 0 | 1 |
| 5 | 42 |
| 10 | 16796 |
| 15 | 9694845 |

* **9.Matrix of Numbers**

Write a program that reads from the console a positive integer number **n** (1 ≤ **n** ≤ 20) and **prints a matrix** like in the examples below. Use two nested loops. Examples:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **n** | **matrix** |  | **n** | **matrix** |  | **n** | **matrix** |
| 2 | 1 2  2 3 | 3 | 1 2 3  2 3 4  3 4 5 | 4 | 1 2 3 4  2 3 4 5  3 4 5 6  4 5 6 7 |

* **10.Odd and Even Product**

You are given **n** integers (given in a single line, separated by a space). Write a program that checks whether the product of the odd elements is equal to the product of the even elements. Elements are counted from **1** to **n**, so the first element is odd, the second is even, etc. Examples:

|  |  |
| --- | --- |
| **numbers** | **result** |
| **2** 1 **1** 6 **3** | yes  product = 6 |
| **3** 10 **4** 6 **5** 1 | yes  product = 60 |
| **4** 3 **2** 5 **2** | no  odd\_product = 16  even\_product = 15 |

* **11.Random Numbers in Given Range**

Write a program that enters 3 integers **n**, **min** and **max** (**min** ≤ **max**) and prints **n** random numbers in the range [**min**...**max**]. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **min** | **max** | **random numbers** |
| 5 | 0 | 1 | 1 0 0 1 1 |
| 10 | 10 | 15 | 12 14 12 15 10 12 14 13 13 11 |

Note that the above output is just an example. Due to randomness, your program most probably will produce different results.

* **12.\* Randomize the Numbers 1…N**

Write a program that enters in integer **n** and prints the numbers 1, 2, …, **n** in random order. Examples:

|  |  |
| --- | --- |
| **n** | **randomized numbers 1…n** |
| 3 | 2 1 3 |
| 10 | 3 4 8 2 6 7 9 1 10 5 |

Note that the above output is just an example. Due to randomness, your program most probably will produce different results. You might need to use [arrays](http://msdn.microsoft.com/en-us/library/aa288453(v=vs.71).aspx).

* **13.Binary to Decimal Number**

Using loops write a program that converts a [binary integer](http://en.wikipedia.org/wiki/Binary_numeral_system) number to its decimal form. The input is entered as **string**. The output should be a variable of type **long**. Do not use the built-in .NET functionality. Examples:

|  |  |
| --- | --- |
| **binary** | **decimal** |
| 0 | 0 |
| 11 | 3 |
| 1010101010101011 | 43691 |
| 1110000110000101100101000000 | 236476736 |

* **14.Decimal to Binary Number**

Using loops write a program that converts an integer number to its [binary representation](http://en.wikipedia.org/wiki/Binary_numeral_system). The input is entered as **long**. The output should be a variable of type **string**. Do not use the built-in .NET functionality. Examples:

|  |  |
| --- | --- |
| **decimal** | **binary** |
| 0 | 0 |
| 3 | 11 |
| 43691 | 1010101010101011 |
| 236476736 | 1110000110000101100101000000 |

* **15.Hexadecimal to Decimal Number**

Using loops write a program that converts a [hexadecimal HYPERLINK "http://en.wikipedia.org/wiki/Hexadecimal" integer](http://en.wikipedia.org/wiki/Hexadecimal) number to its decimal form. The input is entered as **string**. The output should be a variable of type **long**. Do not use the built-in .NET functionality. Examples:

|  |  |
| --- | --- |
| **hexadecimal** | **decimal** |
| FE | 254 |
| 1AE3 | 6883 |
| 4ED528CBB4 | 338583669684 |

* **16.Decimal to Hexadecimal Number**

Using loops write a program that converts an integer number to its [hexadecimal HYPERLINK "http://en.wikipedia.org/wiki/Hexadecimal" representation](http://en.wikipedia.org/wiki/Hexadecimal). The input is entered as **long**. The output should be a variable of type **string**. Do not use the built-in .NET functionality. Examples:

|  |  |
| --- | --- |
| **decimal** | **hexadecimal** |
| 254 | FE |
| 6883 | 1AE3 |
| 338583669684 | 4ED528CBB4 |

* **17.\* Calculate GCD**

Write a program that calculates the [**greatest common divisor**](http://en.wikipedia.org/wiki/Greatest_common_divisor) (**GCD**) of given two integers **a** and **b**. Use the **Euclidean algorithm** (find it in Internet). Examples:

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **GCD(a, b)** |
| 3 | 2 | 1 |
| 60 | 40 | 20 |
| 5 | -15 | 5 |

* **18.\* Trailing Zeroes in N!**

Write a program that calculates with how many zeroes the factorial of a given number **n** has at its end. Your program should work well for very big numbers, e.g. n=100000. Examples:

|  |  |  |
| --- | --- | --- |
| **n** | **trailing zeroes of n!** | **explaination** |
| 10 | 2 | 36288**00** |
| 20 | 4 | 243290200817664**0000** |
| 100000 | 24999 | think why |

* **19.\*\* Spiral Matrix**

Write a program that reads from the console a positive integer number **n** (1 ≤ **n** ≤ 20) and **prints a matrix** holding the numbers from **1** to **n**\***n** in the form of **square spiral** like in the examples below. Examples:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **n** | **matrix** |  | **n** | **matrix** |  | **n** | **matrix** |
| 2 | 1 2  4 3 | 3 | 1 2 3  8 9 4  7 6 5 | 4 | 1 2 3 4  12 13 14 5  11 16 15 6  10 9 8 7 |