Home task №1 (before 11/03/2023)

## General notes.

* In code for all tasks you need to use only those features of the C# language that were discussed during the training at the time the homework was assigned (some exceptions may be specifically specified in the task conditions).
* One task-> one project (\*.**csproj**), possibly in common solution (\*.**sln**).
* Follow coding conventions (here are links):

<https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md>

<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions>

* Use public **git** repository for tasks solutions, one task-> one git branch. **GitHub** is preferable.
* Add **IgorKheidorov** as collaborator for **git** repo
* Present tasks solutions for a review as **pull request** **(PR)** from **task** branch to **master** branch, make trainer the reviewer for each **PR** .
* Check that your solutions in **git** repository **DO NOT** contain binary files (at least no folders like **bin** and **obj**)
* Merge code from **task** **branch** to **master** after PR **approval** by the reviewer (trainer)

## Task 1.1

For numbers in duodecimal numerical system, the symbols **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (ten) and B (eleven)** are used. When starting, the application asks the user to input two integers **a** and **b** (assume that the user enters integers without errors). Then the application displays **all** integers in the range from **a** (inclusive) to **b** (inclusive), which in their duodecimal representation have exactly **two** symbols **A**. Develop a console application that implements the specified functionality.

*Note 1.* In order to transform **string** s to **int** use method **int.Parse(s)**.

*Note 2.* Output required numbers in decimal (not duodecimal) numerical system.

## Task 1.2

Let [10-character ISBN](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%B4%D1%83%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%8B%D0%B9_%D0%BD%D0%BE%D0%BC%D0%B5%D1%80_%D0%BA%D0%BD%D0%B8%D0%B3%D0%B8) is an unique digital code to identify a book, with form: . Digit is a control one, it is calculated according the condition, that expression

(the sum of the products of digits by the weight of their positions) has to be a multiple of 11.

Create an application that prompts the user for a string of 9 digit characters (these are the first nine digits of the ISBN), calculates the check digit, and outputs the resulting ISBN. Do not check the correctness of the user's input - assume that the user does not make errors when entering.

*Note 1.* Сheck «digit» can be equal to 10. In this case use symbol X to denote it.

*Note 2.* You can convert any **a** value to string using **a.ToString()**.

## Task 1.3

When starting, the application asks the user for the number of elements in the integer array, and then in a loop - the elements themselves (of type **int**). After entering the elements, the application outputs the original array. Then a new array is formed and output from the original array according to the principle “each value – only once”. For example, for the array [1, 1, 3, 4, 2, 2, 6, 7, 1], the array [1, 3, 4, 2, 6, 7] is formed. Develop a console application that implements the specified functionality.

*Note 1.* The correctness of the array length may not be controlled.

*Note 2.* Do not use the standard collections and LINQ!