Homework 4 (String Pattern Matching)

**Instructions**

Every assignment needs to provide two main artifacts – a complete working program, along with its full source code, and your solution documentation. The documentation should be in a single text file (e.g. doc), placed inside your Solution folder.

**For this assignment, the documentation will make 10% of the grade, and code will be 90%.**

In the documentation you need to describe the changes you made to the classes and the driver program, to fulfil the assignment requirements.

The software fraction of the homework will be graded according to the following rules:

* existence of compile-time errors not more than 20 %
* existence of run-time errors not more than 50 %
* existence of logical-errors not more than 70 %
* correct program not less than 70 %

The last 30 points for the software part will be given for **program comments**, efficiency, class hierarchies, textual representation, structure, layout, etc.

**NOTE:** If the program is not correct, credits for these properties will not be given!

**NOTE:** An incomplete implementation of the required functionality is considered a logical error.

**Deliverables**

A zipped top-level folder of the Visual Studio solution, containing all of the code, plus the document describing the program.

**NOTE:** You may be asked to demonstrate your solution in person.

**Deadline**

The complete homework (code and documentation) must be uploaded on Canvas no later than Sunday, 25th February, 2024, 23:59.

**NOTE: Late work will be accepted but penalized.**

**Task Description**

Use the code developed for **HW 2**.

Implement the KMP or BM string pattern matching algorithm. If your ID number is odd – implement KMP, otherwise (if even) – implement BM.

**Driver Program:**

Modify the classes’ code (if needed) and write a program to show the items that contain a certain search string (entered by the user) in their name. Your code should traverse only the *Items* collection that contains pointers to objects of both Artifact and Service classes.

After using the above algorithm to find which items contain the string, display their names and GUID, ordered by the number of occurrences of the string, from high to low.