# Homework: Collection Data Structures and Libraries

This document defines the **homework assignments** for the ["Data Structures" course @ Software University](https://softuni.bg/trainings/1147/Data-Structures-June-2015). Please submit a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

## Find Words in File

Write a program that finds a **set of words** (e.g. 1000 words) in a **large** **text** (e.g. 100 MB text file). Print how many times **each word** occurs in the text. Ensure your program works fast enough.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Hello, I am studying C# at SoftUni.  C# is my favorite language.  I like C# and Java.  at at at  2  C#  at | C# -> 3  at -> 4 |

Hint: use Dictionary<string, int>.

## Products in Price Range

Write a program to read a **large collection of products** (name + price) and efficiently **find the first 20 products** in the **price range [a…b]** ordered by price. Test for **500 000 products** and **10 000 price searches**.

Hint: you may use OrderedBag<Product> or OrderedMultiDictionary<int, Product> and **sub-ranges**.

## Implement a Binary Heap

Implement a class PriorityQueue<T> based on the data structure "**binary** **heap**". Do not use any external library!

Hint: learn about ["**binary heaps**" in Wikipedia](https://en.wikipedia.org/wiki/Binary_heap).

## String Editor

You have to implement a string editor that starts from empty string and executes sequence of commands:

* INSERT some\_string position – inserts given string at given position. Print "ERROR" in case of invalid position.
* APPEND some\_string – appends given string at the end of the text.
* DELETE start\_index count – deletes the specified substring. Print "ERROR" in case of invalid substring.
* REPLACE start\_index count some\_string – replaces the specified substring with the specified string. Print "ERROR" in case of invalid substring.
* PRINT – prints the string in the editor.

Ensure your programs runs **efficiently** for tens of thousands of commands.

Hint: use **rope of chars**.