Data Structures & Algorithms III– SCS 2201
String Matching Assignment
Deadline: on or before 22nd July 2022 (11.55 pm)

In this assignment you will implement a program for searching a module catalogue from a given list.

Problem

Write a program that performs the following tasks using a familiar programming language (Java/C/C++/Python):

1) Prompts the user to enter a pattern P to search for, for example "Architecture".
2) It then proceeds by reading the module catalogue stored in the file 'modules.txt', where each line contains the title of one module. Here is a short fragment from the file:

```
…
APL8006 Advanced Landscape Design Studio 2
APL8007 Design Thesis MALAS
APL8008 History of the Designed Landscape
ARC1007 Architectural Design 11
ARC1015 Introduction to Architecture
…
```

   'module.txt' file is attached.

3) For every line in the file, the program checks whether it contains the search pattern P. If yes, it prints out this line. For example, the line corresponding to the `ARC1015` module (see above) contains the string "Architecture" and should be printed. Note: string matching should be case-insensitive, i.e. "Architecture" contains the search string "architecture".
4) When all lines have been processed, the program prints the number of found matches.

See examples below for a clarification. Note: you do not have to follow exactly the same input and output format as shown in the examples. You are also encouraged to experiment by adding other useful features for manipulating the information provided in the module catalogue.

Examples:

1.

```
Enter a search string: Microprocessor
EEE2007 Computer Systems and Microprocessors
EEE2206 Computer Systems and Microprocessors
EEE3010 Microprocessor Control
EEE8022 Microprocessor Systems
Number of matches: 4
```

2.

```
Enter a search string: circuits
Number of matches: 0
```

How to Deliver

- Your solution should contain
  - Compilable (and afterwards runnable) source file(s) of your implementation.
  - Readme file containing details of how your solution can be run: preparing the environment to run your program, location of makefiles, etc.
  - You should convince the examiner, best in a concise and clear manner, that your solution does the job.
    - An explanation: justifying the selection of the string-matching algorithm in a separate file. However, a lengthy discussion of the principles of the selected string-matching algorithm is not needed. Concentrate on your solution and highlight the special points of the code. Salient comments in the source code may also help. However, a general explanation should be in the file, not part of the comments.

You should upload a Zip file with your solution. It is important that you use your index number to name the zip file.

Plagiarism

All programming work must be your own. All forms of plagiarism and cheating (for example downloading programs directly from the internet or copying from another student) are regarded seriously and could result in heavy penalties including failure in the assignment. Under certain circumstances some students may be called for a viva.