Data Structures & Algorithms III– SCS 2201

String Matching Assignment

Deadline: on or before 28th June 2021 (11.55 pm)

You are required to implement a suitable string-matching algorithm to offer the user the ability to find a pattern in a text when the pattern contains "wild-cards". A wild-card is a special "symbol" which is meant to match any letter. For this assignment only, the underscore is considered as a wildcard. For concreteness, it is assumed that the text does not contain the "_"-symbol (the underscore). For illustration, the "pattern" **c_g** can be found in the following "text" **cog**wrgac**cag** two times (at the beginning and at the end).

1. The pattern should be allowed to contain more than one wild-card (i.e., more than one "_" and for example c_ _g, c_g_, c_ _g_).
2. Your algorithm should find all positions where the *pattern* is found.

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. That should be based on two things:
    - **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.
    - **Test-data**: Convince the examiner by preparing some test runs. Provide instructions as to how to execute the tests. As a suggestion: make a small number of tests. Therefore, provide **n** pairs of *Pattern* and *Text*, for instance pattern1.txt, text1.txt, pattern2.txt, text2.txt etc. and produce files patternmatch1.output, patternmatch2.output. Make meaningful test cases (for instance, try also the empty pattern/text).

- **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.