Challenge 1: TRIES

1 Requirements

1.1 Theory Part

Trie is an efficient *information re**Trie**val* data structure. Using *Trie*, searching operation can be brought to optimal limit (key length). Using the code given (**Tries.pdf**) as the reference, students are requested to fulfill the following requirements:

1. Show the time complexity of the following operations (of *Trie*):
    - Adding a word.
    - Removing a word.
    - Searching a word.
    - Searching words which has the same prefix with length *i*.
2. Show the advantages of *Trie* comparing to other data structures (designed for searching) which you have learned: `Binary search tree` and `Hash table`.

1.2 Programming Part

1. Students implement and build a Trie (with words from the given files, including:
    - File *Dic.txt* contains a list of English words sorted ascending. Each word locates on a single line. You have to use this list of words to build the Trie.
    - File *Tries.pdf* contains the source code for operations on the Trie data structure. **You do not need to use source code from this file**.

2. With the built *trie* (containing the provided English words), implement a program to generate a list of *valid English words*[1] which have letters from a given character list. (*Note: Valid English words do not require to have all of provided letters but must have at least 3 letters*).

    - **Input:** List of letters use for creating valid English words. These letters must be on the same line and satisfied the following requirements:
        - Being in lowercase, not in CAPITAL.
        - Being sorted ascending in lexicographic order, separated by a single space `" "`.
        - May appear multiple times.
    - **Output:**
        - The 1st line: An integer `N` indicates the number of created words.
        - Next `N` lines: each line contains a created word.
    - Example:

Input	**Output**
a c e p	6 ace ape cap cape pace pea

[1] valid English words are words existed in the given dictionary file.

2 Regulations and Evaluations

2.1 Regulations

- This challenge requires a group of 4 students.
- The submission file must be in the following format:

  [**StudentID1-StudentID2-StudentID3-StudentID4.rar/.zip**]

  Example:

  - Given the student codes: *21120666 - 23120888 - 23120991 - 21120999*.

  → **The name of submission file is**:

  21120666-21120999-23120888-23120991.zip/rar.

  This folder contains:

  - The report file must be presented as a document [**report.pdf**]. This file presented research answers from **Theory part** and the solution of problems from **Programming part**.
    - ∗ Information (Name, Student's ID) must be provided on the first page of your report.
    - ∗ The report file should be structured, logical, clear, coherent, and answer directly to the question. The length of the submission should not exceed 15 pages for the document file.
  - The source code must follow the requirements in **Programming part**. The main program [**main.cpp**] should be clear, logical and commented.

2.2 Evaluation

- Submission with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board of the the program for further action.

END