PA2

You have to **re-register into Groups**, PA2_Group_..., again max 3 students per group.

This is the last assignment with limitations on what can be used.

You can **NOT** use ***std::string*** or any STL data structure, e.g. ***std::vector***, ***std::set***, ***std::list***

You can include *<string.h>* and use **strlen**, **strcat**, **strcpy** and other operations that use ***char**** (or **char[]**) directly.

*Students are meant to practice using pointers and arrays directly, both when working with the strings (***char****) and lists of other stuff*.

Also learn to seed and generate random numbers in C++, to make word choice and jumbles more interesting.

Handing in

- Hand in a single ZIP, including:
    - all code files (.cpp and .h)
        - *hopefully more than just a single .cpp as* **classes and good code organization** *can really help*
    - files and scripts needed to run
        - *in PA2* ***probably a wordbank file*** *but not much more here*
    - README.txt in which is
        - What version of the assignment was solved and should be graded
        - What needs to be written in the terminal to compile and run the program
        - Anything else a student wishes to say (Canvas comments will **not** be delivered to the graders)
- I don't wish to ***force all-nighters upon my students*** but I feel it is their choice, so I will **not reduce** grades if submissions are LATE, as long as they are there when I wake up the next morning and collect the assignment solutions from Canvas.

I will update this description if the need arises. Right now it is rather open to interpretation, but I wanted to get it published more than I wanted to get it polished :)

Use the power of classes and pointers to your advantage throughout this assignments.

Version A (60%)

Take a word from a word list, stored in text file (decide on the file format yourselves and make the wordbank file).

Scramble the letters of the word.

Show the scrambled word.

User can guess the word.

Program continues until user doesn't want to continue anymore.

Version B (80%)

Everything in Version A plus:

User can ask for a hint.

The word is shown with dashes, but one random letter is in the correct place.

For each hint, one correct letter is added.

The user should also see the scrambled word, in addition to the correct letters.

User starts with 10 points and loses one point for each hint.

Game is either:

How many words can you guess before you have 0 points.

or:

How many points do you have left after 7 words.

5% bonus on Version B (included in Version C):

The time elapsed since the jumble is shown until the answer is correct is also a factor, deducting points. Find a way to make this work in the overall point calculation. You can re-invent the entire point calculation if you want.

Version C (100%)

Everything in Version B plus:

Add time as a factor (*see 5% bonus on v.B*)

Add highscore table *with at least the score for a game/session and the name or initials of the player*.

For the highscores to be interesting add more ways to calculate scores. Longer streaks, speed bonuses, etc.

Store highscores in file.

Allow user to choose to see top 5 scores or full table.

5-10% bonus on any version

Crossword jumble

select 3 or more words from the word bank.

Line them up horizontal and vertical so that each word has a common letter with at least one other word. Print them to the screen using dashes for the letters (or hash or whatever looks cool and is clear). Then print jumbles for each word.

Allow the user to input the solution in a clear way, one at a time. Words probably need to labelled with a number or code, for the user to be able to enter them in the correct order or choose which one to enter. If a correct word is entered, display it in the crossword, while the rest is still dashes or such.