CS300/CS500
Advanced Object-Oriented Programming with C++

Lab #5:

Part I (5 points): Create a function called emailValidator. This function should take a string and return true if the string is a valid email address, otherwise it should return false. An example address email address follows the form:

firstName.lastName@domain.domainExtension.

- **firstName -** can be no larger than 20 characters, and no smaller than 1 character. It can contain numbers, uppercase letters, and lowercase letters
- **lastName** - must be at least 1 character long. Can contain the symbols **!**, **$**, and **&** (commas not included), as well as uppercase and lowercase letters.
- **@** - must exist between **lastName** and **domain**
- **domain** - can be no larger than 20 characters, and no smaller than 1 character. Can only contain lowercase letters.
- **.** - must exist between **domain** and **domainExtension**
- **domainExtension** - can be between 1 and 3 characters long, and must contain only lowercase letters.
- There cannot exist any spaces in the given email

Create an emailValidator.h file and an emalValidator.cpp file. Include a prototype for emailValidator function in the emailValidator.h file, and the definition for this function in the emailValidator.cpp file. Your function prototype should look like this:

bool emailValidator(std::string s);

You can name the string variable whatever you like

Examples

Input: "joe.Mack@gmail.com"
Output: true

Input: "Henry1999.Tuttle!@uab.edu"
Output: true

Input: "jasonIsTheCoolestManEver.Horatio Nelson@Outlook.edu"
Output: false (*firstName is too long, no spaces allowed*)

Part II (5 points): One of your friends is a professor, and has written a recommendation letter for one of their students. Unfortunately they have a hard time spelling the student's name correctly. The student's name is Barthalamuel, but the professor has spelled it as "Barthalamule", "barthle", "bartalAmul" and other such variations.

You've been tasked with making a program that corrects your friends' spelling. Create a function called **spellCheck** that takes the recommendation letter as a string, as well as another string that represents the correct spelling of the students name (this was requested so the professor could use the same recommendation letter for multiple students). The function should then return the recommendation letter as a string with all instances of the misspelled name replaced with the given corrected students name.

Note: You've noticed a pattern in the way your friend spells the students name. The professor always starts the name with "bart" or "Bart" and then continues with a random string of uppercase or lowercase letters that always ends with a space.

Create a spellCheck.h file and a spellCheck.cpp file. Include a prototype for spellCheck function in the spellCheck.h file, and the definition for this function in the spellCheck.cpp file. Your function prototype should look like this:

std::string spellCheck(std::string recLetter, std::string studentName)

You can name the string variables whatever you like

Examples:

Input: "Barthalamule is a fantastic Student. barthle once raised a clutch of baby hawks to adulthood. There was never a challenge that bartksafdjksdfNASmwik couldn't solve.", "Barthalamuel"

Output: "Barthalamuel is a fantastic Student. Barthalamuel once raised a clutch of baby hawks to adulthood. There was never a challenge that Barthalamuel couldn't solve."

Submission: Push all files to a public github repo that only contains the files for this lab. Make sure there is no main function in any of these files.