Lab 6 Iterators

Implement iterators for the String class you made in Lab 3. Please read about iterators for STL containers will work. STL stands for the C++ Standard Template Library.

The purpose of an iterator is to work in almost the same way as a pointer. Read more in C++ primer page 106 et seq.; see also [http://en.cppreference.com/w/cpp/string](https://www.microsofttranslator.com/bv.aspx?from=sv&to=en&a=http%3A%2F%2Fen.cppreference.com%2Fw%2Fcpp%2Fstring).

Note to make a correct implementation of iterators that matches perfectly with the STL system is quite a lot of code. We confine ourselves to implement what is needed to make std:: stable\_sort work.

An iterator iterates over the elements in a container object. In our case it is the String objects from the previous lab we will iterate over. Please note that we need to add some things in the String class in order for it to work.

Task:

You should make all two/four types of iterators:

* iterator
* const\_iterator (VG)
* reverse\_iterator
* const\_reverse\_iterator (VG)

In the String class add: (const versions only for VG).

* typdefs for iterator, const\_iterator, reverse\_iterator and const\_reverse\_iterator
* fucntions begin, end, cbegin, cend, rbegin, rend, crbegin and crend.

Iterators should be able to do:

* \*it, --it, ++it, it++, (it+i), (it-it), it[i], ==, != och <
* default constructor, copy constructor and assignment (=)
* -iteratorn should contain all needed typedef (use std:: iterator)

A tip: the program below prints "jeh"

String ("Hello");

for (String:: reverse\_iterator rbegin it = p. (); it! = p. rend (); ++ it)  
cout << \* it;

**Please note that you are not allowed to iterator adapters from STL to solve your task.**

**The test application in Main .cpp**

Please note that the test application that is available in Main .cpp just is a help and neither completely nor guaranteed to be completely accurate. It is possible to test application running flawlessly though your solution is incorrect. It is also possible, but not likely, that your solution is correct though the test program does not run/compile without error. The program consists of several files. See the files on ITSL. VG.h defines if you try for VG.

# Requirements for G: Do the above, except const-iterators

# Requirements for VG:

Do all the iterators. Your software must also have all const in the right place, be effective and devoid of unnecessary code duplication. This means that you have to make same form of template class BaseItertor and then use it by inheritance or typdef.