4.1 Merge sort template

Due 28 Sep 2020 by 23:59 **Points** 0 **Submitting** an external tool

For this assignment, you will implement the *merge sort* algorithm as a function template. You can start from the example code in the zyBook and translate it such that is **does not use** the following items:

- arrays
- the [] operator on arrays or vectors
- pointers and dynamic allocation (no "*new int[mergedsize]*")

But you can also write your own *merge sort*, if you prefer. In any case, your code must only use vectors, no arrays.

To give this assignment another twist, we want you to write *merge sort* as a template function that works independently of the type of elements to be sorted. For this, we want you to write a file **mergesort.h** that implements a template function according to the following declaration:

```
template <typename T> void mergeSort(std::vector<T>& data,
                                     unsigned int firstToSort,
                                     unsigned int lastToSort);
```

mergeSort takes a vector *data* of unsorted values. An invocation of *mergeSort* sorts the elements from *data.at(firstToSort)* up to *data.at(lastToSort)*.

For testing your template function, you can use the following two main programs that are identical to each other, except for the type of vector elements to be sorted. These programs are already present on CodeGrade.

- **mergesortdouble.cpp** **(https://canvas.vu.nl/courses/50259/files/2701746/download?download_frd=1)**
- **mergesortstring.cpp** **(https://canvas.vu.nl/courses/50259/files/2701747/download?download_frd=1)**

You may need to use the option **-std=c++14** when compiling these programs with *g++*.

This tool was successfully loaded in a new browser window. Reload the page to access the tool again.