Creative Software Programming, Assignment 7-1

Handed out: Oct 20, 2021

Due: 23:59, Oct 20, 2021 **(NO SCORE for late submissions!)**

- *Only files submitted by **git push to this course project at https://hconnect.hanyang.ac.kr** (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git) will be scored.*
- *Place your files under the directory structure **<Assignment name>/<Problem no.>/<your files>** just like the following example.*
  - ***DO NOT push CMake intermediate output files** (CMakeCache.txt, cmake_install.cmake, Makefile, CMakeFiles/*).*
  - ***DO NOT use cmake_minimum_required()** command in CMakeLists.txt.*

```
+ 2020_ITE0000_2019000001
  + 2-1/
    + 1/
      - 1.cpp
      - CMakeLists.txt
    + 2/
      - 2.cpp
      - CMakeLists.txt
    + …
```

- *The submission time is determined not when the commit is made **but when the git push is made**.*
- *Your files must be committed to the **master branch**. Otherwise, it will not be scored.*
- *Your program should output correct results even for inputs other than those used in the example.*
- *Basically, assignments are scored based on the output results. If it is not possible to check whether a requirement is implemented because the output is not correct, no score is given for the requirement, even if it is implemented internally. However, even if the output result is correct, no score is given for a requirement if the internal implementation does not satisfy the requirement.*

1. Write a program for a sorted number array.
    A. Implement class SortedArray in the code skeleton.
    B. Take an arbitrary number of integers from the user and add them to a SortedArray instance.
    C. Process user commands as described in the Input.
    D. This program should take user input repeatedly
    E. Note that
        i. DO NOT write your own code for sorting and getting min / max. Use STL functions instead.
    F. Input:
        i. First, integer numbers to build a sorted array.
        ii. 'ascend' – Print out the numbers in ascending order.
        iii. 'descend' – Print out the numbers in descending order.
        iv. 'max' – Print out the maximum number among the numbers.
        v. 'min' – Print out the minimum number among the numbers.
        vi. 'quit' – Quit the program.
    G. Output: The result of each command.
    H. Files to submit:
        i. main.cpp – main() must be in this file.
        ii. sorted.h – Just copy the following code skeleton.
        iii. sorted.cpp – Implements SortedArray member functions.
        iv. A CMakeLists.txt to generate the executable

```
$ ./sorted_array
9 3 6 2 7
ascend
2 3 6 7 9
descend
9 7 6 3 2
max
9
min
2
```

```
10 3
ascend
2 3 3 6 7 9 10
quit
$
```

Code skeleton:

```
class SortedArray
{
public:
	SortedArray();
	~SortedArray();

	void AddNumber(int num);

	 std::vector<int> GetSortedAscending();
	 std::vector<int> GetSortedDescending();
	 int GetMax();
	 int GetMin();

private:
	 std::vector<int> numbers_;
};
```

2. Write a program for checking palindrome.
   A. Implement class CharList in the code skeleton.
   B. This program should take user input repeatedly.
   C. Input:
      i. Check if the list is a palindrome.
         1. Tip: It is convenient to use std::boolalpha to print out a boolean value as a string "true" or "false". (ex. std::cout << std::boolalpha << (a Boolean value) << std::endl;)
      ii. 'quit' – Quit the program.
   D. Output: The result of each command.
   E. Files to submit:
      i. main.cpp – main() must be in this file.
      ii. charlist.h – Just copy the following code skeleton.
      iii. charlist.cpp – Implements CharList member functions.
      iv. A CMakeLists.txt to generate the executable

```
$ ./palindrome
```

```
apple
false
racecar
true
quit
$
```

Code skeleton:

```
class CharList {
	public:
		CharList (const char* str);
		~ CharList ();

		bool CheckPalindrome();
		std::list<char> GetAll();

	private:
		std::list<char> mystring_;
}
```