Creative Software Programming, Assignment 5-1

Handed out: Oct 6, 2021

Due: 23:59, Oct 12, 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 that works as follows.
   A. Take a string (s) and an integer (n) as command line arguments.
   B. Print out the string s n times.
   C. Input: A string and an integer (as command line arguments)
   D. Output: Strings n times
   E. Files to submit:
      i. A C++ source file
      ii. A **CMakeLists.txt** to generate the executable

```
$./command_argument asdf 3
asdf
asdf
asdf
```

2. Write a program that works as follows.
   A. Take arbitrary number of strings and integers as command line arguments.
      i. Assume that each input integer is always bigger than 0.
      ii. Ignore program name(.exe) among the arguments.
   B. Concatenate all the strings and sum all the integers.
   C. Print out each result.
   D. Note that:
      i. atoi() returns string → integer conversion, 0 if impossible.
   E. Input: Strings and Integers (as command line arguments)
   F. Output: Concatenation and summation of each type
   G. Files to submit:
      i. A C++ source file
      ii. A **CMakeLists.txt** to generate the executable

```
$ ./cmdarg 1 20 aaa bbb
aaabbb
21
```

```
$ ./cmdarg aa 20 10 bbb dd 100 ee
aabbbddee
130
```

3. Write a program that works as follows.
   A. Take arbitrary number of integers as command line arguments.
   B. Create an array of length N (the number of input integers), and fill the array with input integers.
   C. Find the min and max values in the array using getMinMax() in the following form.
      i. void getMinMax(int* arr, int len, int& min, int& max);
   D. Print out the min and max values.
   E. Note that
      i. You must use new [ ] operator to allocate the array.
      ii. Do not forget to free the memory by using delete[] operator after using the array.
      iii. Only <iostream>, <cstdlib> are allowed to include.
   F. Input: Integer values (as command line arguments)
   G. Output: Min and max values
   H. Files to submit:
      i. **main.cpp** – main() must be in this file.
      ii. **minmax.cpp, minmax.h** – getMinMax() must be in these files.
      iii. A **CMakeLists.txt** to generate the executable

```
$ ./dynamic_min_max 1 2 2 -1 -1
min: -1
max: 2
```