CMPE-50 Object-Oriented Concepts and Methodologies, Tarng, Spring 2021
Homework #5

Due: 5/16/2021 Sunday midnight

The submission of the homework should be the cpp files with the output in the code comment. Each problem needs to have a complete program, meaning, it needs to contain the main() function and some test code and data. Therefore, you need to submit six cpp files in total. Do not zip the files. Name the files in the following way: CMPE50-HW-5-1.cpp, CMPE50-HW-5-2.cpp, etc. If the solution includes some input files, also submit the input files.

Chapters: 8.3 (Vectors), 17 (Templates)

Total 40 points

1. [10 pts] (Chapter 8.3: Vector)
   Write a sorting function that is similar to Display 7.12 in Chapter 7 except that it has an argument for a vector of ints rather than an array.
   - Replace “int sample_array[10];” with “vector<int> sample_array;”
   - Replace function prototype “int a[]” with “vector<int> &a”
   - Replace “a[index] = next;” with “a.push_back(next);”

   In addition, remove the parameter like number_used as in Display 7.12, since a vector can determine the number used with the member function size(). This sort function will have only one parameter, which will be of a vector type.

   (Note that the Display 7.12 source code is available on Canvas: SavitchCPP9_SourceCode.zip. The majority of the code is in 7-12.cpp and one function is provided in 7-09.cpp.)

2. [10] pts] (Chapter 8.3: Vector, based on Program Project 8-15)
   Given the following header:
   ```
   vector<string> split(string target, string deliemiter);
   ```
   implement the function split so that it returns a vector of the strings in target that are separated by the string delimiter. For example:
   ```
   split("10,20,30", ",")
   ```
   should return a vector with the strings “10”, “20”, and “30”. Similarly,
   ```
   split("do re mi fa so la ti do", " ")
   ```
   should return a vector with the strings “do”, “re”, “mi”, “fa”, “so”, “la”, “ti”, and “do”.

3. [10 pts] (Chapter 17: Function Template)
   Write a search function template that has three parameters. The first one is a partially filled array, the second one is the number of entries in the array, and the thrid argument is a value of the base type of the array. If the value is in the partially filled array, then the functionr returns the index of the first occurrnce of that value, otherwise, the function returns -1. The base type of the array is a type parameter.
   - You should start by assuming the base type of the array is int. Write a search function for a partially filled array of int.
     For example, the following is the function prototype of search for an interger array:
     ```
     int search(int arr[], int number_used, int target);
     ```
   - Make sure this search function is working properly. Then convert the search function into a template.

   Write a test main function and test the search function template by using int, char, and string as the type parameter.

4. [10 pts] (Chapter 17: Class Template)
   Write a template-based class `Set` that implements a set of items. A `Set` is a collection of items in which no item occurs more than once. Internally, you may represent the set using the data structure of your choice (for example, vector or arrays). However, the class should externally support the following member functions:
   - `add`: Add a new item to the set. If the item is already in the set then nothing happens.
   - `remove`: Remove an item from the set.
   - `size`: Return the number of items in the set.
   - `contain`: Determine if an item is a member of the set.
   - overloaded `operator <<`: Print out the items