Algorithms and Data Structures I - Fall 2018 - Programming Assignment 2

Directions: This assignment will be demoed in lab on Thursday, October 25th or Friday, October 26th.

Lab 2: Sorting Algorithms Assignment Details: In this lab, you will implement several different sorting algorithms using C++, namely:

- Insertion Sort
- Selection Sort
- Merge Sort

You may base your code on the implementation of bubble sort in C++ listed at the bottom of this lab assignment.

Your program should accept on the command line the name of a sorting algorithm along with a list of integers. For example:

```
./a.out bubble 12 3 -9 3 100 0
```

Bubble Sort Implementation

```
#include <iostream>

void bubble_sort(int a[], int n)
{
  int temp;
  for(int i = 0; i < n; i++) {
    for(int j = 0; j < n - i; j++) {
      if(a[j] > a[j+1]) {
        temp = a[j+1];
        a[j+1] = a[j];
        a[j] = temp;
      }
    }
  }
}

// print an array with length n to standard output
void print_array(int a[], int n)
{
  for(int i = 0; i < n; i++) {
    std::cout << a[i] << " ";
  }
  std::cout << std::endl;
}

int main(int argc, char* argv[]) {

  std::string sort_type = std::string(argv[1]);
```

```
  int n = argc - 2;   // do not count executable name or first argument
  int* arr = new int[n];
  for (int i = 0; i < n; i++) {
    arr[i] = atoi(argv[i+2]);
  }

  std::cout << "Original list: ";
  print_array(arr, n);

  if (sort_type == "bubble") {
    bubble_sort(arr, n);
    std::cout << "Sorted list: ";
    print_array(arr, n);
  } else {
    std::cout << "Sorry, " << sort_type << " sort is not implemented!" << std::endl;
  }

  delete [] arr;
}
```