SCS214: Data Structures
Assignment-2

Instructions

1- Students will form teams of **2** students (**from the same lab group).**
2- Deadline of submission is **Sunday April 3rd at 11:55 pm.**
3- Submission will be on Blackboard.
4- No late submission is allowed.
5- No submission through e-mails.
6- **Please follow the Submission Notes found below**
7- **In case of Cheating you will get a negative grade whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.**
8- You have to write clean code and follow a good coding style including choosing meaningful variable names.

Task

Assuming you have a class Student

```
class Student
{
    string name;
    string id;
    double gpa;
    public:
      Student(string, string, double);
      void print();
};
```

1- Implement the class constructor and print member function.
2- Overload the operator < such that it compares the names of 2 student objects.
3- Save the class declaration and implementation in a single header file named as Student.h
4- Include your Student.h in the main file.
5- Now implement as **template functions**:
   - A quadratic sorting algorithm (bubble, selection or insertion)
   - The sub-quadratic algorithm Shell Sort with Hibbard's increments (look it up).
   - Another sub-quadratic algorithm (quick or merge sort)

SCS214: Data Structures
Assignment-2

6- Read into 3 dynamic arrays of student objects (a1, a2 & a3) from a file named students.txt which will have the number of students followed by their info as follows:

```
5
Sara Ahmed
64387
3.2
Hany Ali
36572
2.5
Hala Samir
957693
3.7
Kamal Mohamed
46598
2.2
Laila Fayed
4555467
3.9
```

7- Sort the 3 arrays , each using one of the 3 sorting algorithms you implemented in step 5. For example a1 will be sorted using selection sort, a2 will be sorted using shell sort, & a3 will be sorted using quick sort.

8- Each of the 3 sorted arrays should be saved in a separate txt file with the name of the sorting algorithm. So you will have 3 txt files named ,**for ex., Bubble.txt, Shell.txt, and Merge.txt**.

9- You should count comparisons for each of the sorting algorithms you implemented such that the number of comparisons made by each sorting algorithm will be written as the first line in the txt file that contains the array sorted by this algorithm. An example text file follows.

10-Implement a binary search algorithm to work on any of your sorted students arrays, such that given a student's name the search returns its index or -1 if it's not found.
A binary search has better complexity (O(log N)) than a linear search (O(N)).
You can look up the code of binary search online.

```
Number of comparisons = 10
Hala Samir
957693
3.7
Hany Ali
36572
2.5
Kamal Mohamed
46598
2.2
Laila Fayed
4555467
3.9
Sara Ahmed
64387
3.2
```

Bubble.txt

11-In main you should :

- Read from students.txt to fill the 3 arrays
- Call each of the sorting algorithms on each of the arrays
- Save the results to 3 text files, one for each sort

- Then display the following menu, which will change depending on the algorithms you chose to implement.
- For the search option, the user will enter a student's name and if it's found the whole student record will be shown, otherwise display an appropriate message.

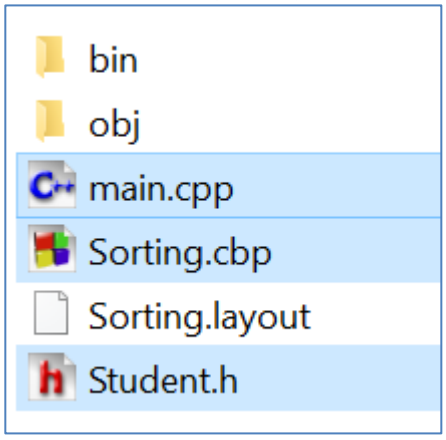
```
1- Show number of comparisons and sorted array of selection sort
2- Show number of comparisons and sorted array of shell sort
3- Show number of comparisons and sorted array of merge sort
4- Search for a student by name
5- Exit
```

Note that the 3 sorting algorithms will be called and their corresponding text files will be generated before displaying the menu above. The above menu is only for the convenience of the user to show the sort results, and for the search functionality.

Submission Notes:

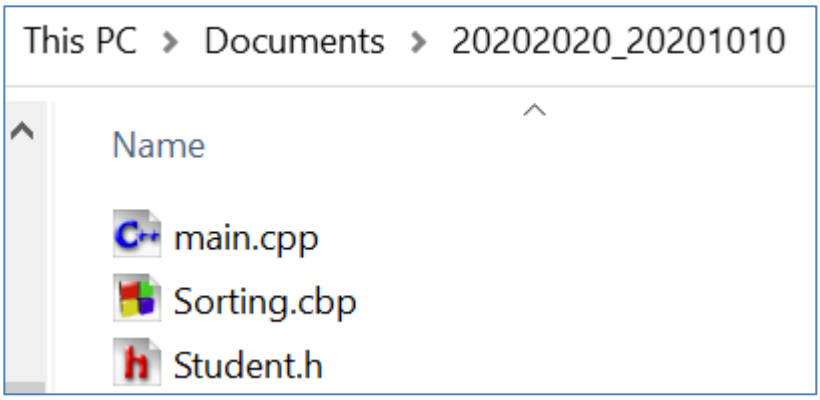
You will have 2 code files in your project: student.h and main.cpp, in addition to the project file (extension cbp). Please copy the 3 files into a new folder named by your ids, then compress it into a zip file. We don't need any other files than those 3.

This is the original project folder:

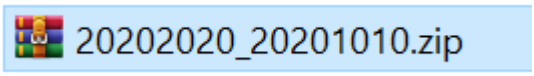


Copy the 3 files from the original folder into a new folder with your ids as a name:

SCS214: Data Structures
Assignment-2



Compress your folder into a zip file:



Grading Info:

Student Class	**5**
Quadratic Sorting Alg.	**20**
Shell Sort	**25**
Sub-Quadratic Soring Alg.	**20**
Use Templates	**10**
Read from file into Arrays	**5**
Write Sorted Arrays into files	**5**
Binary Search	**10**
Number of comparisons	**10**
Separate header file	**5**
Correct Naming/ Submission	**15**
Total	**130**