Programming Assignment 7
Sorting
Due Date : Wednesday April 13th , 2016 - @
3:30 pm for section 2 & 5:00 pm for section 3

Problem:

Write a C++ program that does the following :

Using a random number generator, create an array of 100,000 elements of type integer then perform the following :

1. Display the first 20 numbers.
2. Search for an element in the array using **sequential search** and at the end display number of comparisons it makes.
3. Sort the same array using **Bubble** Sort and at the end display number of exchanges it makes.
4. Sort the array using **Selection** Sort and at the end display number of exchanges it makes.
5. Sort the array using **insertion** Sort and at the end display number of insertions it makes.
6. Sort the same array using **Quick** Sort and at the end display the number of recursion calls it makes.
7. Search for an element in the array using sequential search after the sort and display number of comparisons it makes.
8. Search for an element in the array using **Binary search** and display number of comparisons it makes.
9. For each of the preceding steps (2 thru 8), calculate and print the CPU time before each step starts and after each completed step then calculate actual CPU time for the completion of each step.

NOTES:

- Just one .cpp file with 8 individual functions plus main for testing.
- Do not use any global variables.
- Use the array size as a const variable.
- Do not use any sort library that is available with CodeBlocks or any other IDE.

Style Guidelines:

At the beginning of your program (and **before** the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//      Roster Number / s  :         xxxxxxxxx
//
//      Author / s : (Your name  here!!)
//      Due Date  :
//      Programming Assignment Number  7
//
//      Spring   2016   -   CS 3358    -    Your Section  Number
//
//      Instructor:    Husain Gholoom.
//
//       <Brief description of the  purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. weightInPounds).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. TX_STATE_SALES_TAX)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int  gender;    // integer value for the gender,
                        // 1 = Male , 2 = Female ,
```

Rules : In order to get a full mark :

1. Your program **must compile** and run. We will test your program using Codeblock or Eclipse versions that are available in our CS labs when testing this program.
2. Your program must be **documented according the style above . See the website for the sample programming style program.**
3. Must **use** at least **8** functions **(prototypes and definitions) for all sorting and other functions of this program.**
4. You must use the appropriate libraries in writing this program.
5. Must properly format the output . Sample is provided .
6. You must name your program as :

   - **LastName_FirstName_PG7_SS.cpp**

   Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like : Gholoom_Husain_PG6.cpp (not .cbp)

7. Everyone must upload the electronic version of the program no later than the starting of class time on the due date. **No late assignments will be accepted.** **DO NOT** send your assignment solution via email. **Group members must upload identical copy of the assignment.**

   To upload your program , go to the CS department's website, click on resources , then select homework upload.

8. You must **also** turn in hard copy of your source code no later than the starting of class time on the due date . Should the hard copy consist of more than one page , then , the hard copy must be **stapled**. If you are unable to turn in a printout during class, you can take the program to the computer science department and hand it to the front desk personal (Comal 211) before the deadline. Make sure that the front office stamps the program. Make sure that include the date and time. Finally ,make sure that they place the program in my mailbox. **Only one copy per group.**

   DO NOT slide your program under my office door – It will **NOT** be accepted

9. **Violating any item from the above rules will result in Grade ZERO for the entire assignment.**

   NO EXCEPTIONS.

Sorting Benchmark

Using a random number generator, we are creating an array of 100,000 elements of type integer then performing the following :

1. Displaying the first 20 numbers.
2. Searching for an element in the array using **sequential search** and at the end displaying number of comparisons it makes.
3. Sorting the same array using **Bubble** Sort and at the end displaying number of exchanges it makes.
4. Sorting the array using **Selection** Sort and at the end displaying number of exchanges it makes.
5. Sorting the array using **insertion** Sort and at the end displaying number of insertions it makes.
6. Sorting the same array using **Quick** Sort and at the end displaying the number of recursion calls it makes.
7. Searching for an element in the array using sequential search after the sort is completed and displaying number of comparisons it makes.
8. Searching for an element in the array using **Binary search** and displaying number of comparisons it makes.
9. For each of the preceding steps (2 thru 8), calculating and printing the CPU time before each step starts and after each completed step then calculating actual CPU time for the completion of each step.

Output : -

Sequential Search
Searching for 12345

12345 Was Not found.
Start Time :
End Time :
Actual CPU Clock time :
Number of Comparisons :

Bubble Sort

Start Time :
End Time :
Actual CPU Clock time :
Number of Exchanges :

.
.

Benchmark Algorithm Implemented by : First Name , Last Name
April 2016