CS 161B: Programming and Problem Solving II

Final Exam - Citizen’s Database

Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Purpose

The purpose of this final exam is to test your knowledge of the concepts learned in this course:

- Create a user-defined data type called a struct
- Create an array of structs
- Use C-strings for all string data types
- Shift and insert data in the right position in an array

Task

- ❏ Download the final.zip file and extract the files into a folder of your own on your computer.
- ❏ You can modify the provided `person.h, person.cpp,` and `main.cpp` per the instructions in this document. You are not allowed to use <string>, <vector> or any other header files in STL.
- ❏ Open the Algorithmic Design Document, make a copy, and follow the steps to create your algorithm, **for the function you are going to write.**
- ❏ For this test, you will need to create the following function. First add the function prototype to `person.h,` then put the function implementation in `person.cpp` and finally invoke/test the function in `main.cpp`. Please label your output clearly. E.g. "After adding a person, the list is: "
    - ❏ Create a function to read a person's information and insert into the list at a given position. The function returns true if the insertion is successful and it returns false if the array is out of capacity.
    - ❏ `bool addPerson(PersonType list[], int &count);`
    - ❏ Sample code in `person.cpp`

      ```
      PersonType aPerson;
      //read in person name, person citizenship, and age and
      populate aPerson
      //e.g strcpy(aPerson.name, tempName)
      //read position to insert inside the function - see sample
      //run
      //position could be read from the user or you could set a
      //number that is not more than count. If you read from the
      //user, check to make sure position is not > than count.
      //shift and insert aPerson in the right position
      ```

    - ❏ Sample test code in `main.cpp`

      ```
      if(addPerson(list, count) == true)
      {output list}
      ```

- ❏ You must express your algorithm as **pseudocode** for the function(s) you are going to be writing.
- ❏ You must be able to read cstrings with spaces. You must use cin.getline() to read strings.
- ❏ You must not replace any existing values in the list. You must shift and insert in the right position. See zybooks Lab 11.9 for reference.
- ❏ You must be able to add at the beginning of the list and at the end - if your list has 4 elements, users must be able to insert at position 0 through 4.
- ❏ See Sample Runs below and test your code multiple times adding users at the beginning and at the end.
- ❏ You must make sure the position is within count.

❏ You must check for unreasonable age (for example: age < 1 and age > 100 can be unreasonable).

❏ Include a welcome and goodbye message. (See sample run below).

❏ Must use all the given function prototypes under Task exactly as is. Function Prototypes and implementations must be written in the appropriate files.

❏ Do not add header comments for this exam, but you must have function comments for the function you are writing.

❏ Do not use containers of any sort or any vectors for this program. Use only the concepts we have learned so far.

Criteria for Success

❏ Test your program using the following sample runs, making sure you get the same output when using the given inputs (in `blue`):

```
Welcome to my Citizen's Database.

Here is your list so far:

Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52

Enter your name:
Navid
Enter your citizenship:
Ecuador
Enter your age:
34
Enter position number:
2

After adding a person, the list is:

Gayathri;USA;22
Stephanie;USA;27
Navid;Ecuador;34
Priya;India;34
Ahmed;Nigeria;52

Thank you for using my Citizen Database!!
```

```
Welcome to my Citizen's Database.

Here is your list so far:
```

```
Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52


Enter your name:
Navid
Enter your citizenship:
Ecuador
Enter your age:
34
Enter position number:
9

Error! Invalid position.

Thank you for using my Citizen Database!!
```

```
Welcome to my Citizen’s Database.

Here is your list so far:

Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52


Enter your name:
Lucy
Enter your citizenship:
Ecuador
Enter your age:
34
Enter position number:
0

After adding a person, the list is:

Lucy;Ecuador;34
Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52


Thank you for using my Citizen Database!!
```

```
Welcome to my Citizen’s Database.
```

```
Here is your list so far:

Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52


Enter your name:
Arely
Enter your citizenship:
Mexico
Enter your age:
45
Enter position number:
4

After adding a person, the list is:

Gayathri;USA;22
Stephanie;USA;27
Priya;India;34
Ahmed;Nigeria;52
Arely;Mexico;45


Thank you for using my Citizen Database!!
```

- ❑ Complete all sections of your Algorithmic Design Document.
- ❑ **Follow these coding constructs:**
    - ❑ You must express your algorithm as **pseudocode** for the function(s) you are going to be writing.
    - ❑ You must be able to read cstrings with spaces. You must use cin.getline() to read strings.
    - ❑ You must not replace any existing values in the list. You must shift and insert in the right position. See zybooks Lab 11.9 for reference.
    - ❑ You must be able to add at the beginning of the list and at the end - if your list has 4 elements, users must be able to insert at position 0 through 4.
    - ❑ See Sample Runs below and test your code multiple times adding users at the beginning and at the end.
    - ❑ You must make sure the position is within count.
    - ❑ You must check for unreasonable age (for example: age < 1 and age > 100 can be unreasonable).
    - ❑ Include a welcome and goodbye message. (See sample run).

    - ❑ Must use all the given function prototypes under Task exactly as is. Function Prototypes and implementations must be written in the appropriate files.
    - ❑ Do not add header comments for this exam, but you must have function comments for the function you are writing.
    - ❑ Do not use containers of any sort or any vectors for this program. Use only the concepts we have learned so far.
- ❑ Please open and compare your work with the grading rubric before submitting.
- ❑ Remember to follow all style guidelines.
- ❑ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `fin.pdf`. Upload to the D2L assignment for feedback.
- ❑ Create a zip file of all your files (*.h and *.cpp and any *.txt file) or upload all files individually to the D2L assignment for feedback.
- ❑ Do your own work. Consult the syllabus for more information about academic integrity.