Lab 03 Specification – Exploring Low End Programming in C
Due (via your git repo) no later than 2 PM, Friday, 17th September 2021.
50 points

Lab Goals

- Exploring the implementation of Arrays.
- Exploring the implementation of Pointers and Strings.
- Exploring File System and use of Pointers to access Files.

Learning Assignment

If not done previously, it is strongly recommended to read all of the relevant ”GitHub Guides”, available at the following website:
`https://guides.github.com/`
that explains how to use many of the features that GitHub provides. This reading assignment is useful to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, it is also recommended to do the reading assignment from the section of the course textbook outlined below:

- **KR chapters 1 and 2**

Assignment Details

Now that we have discussed some more of C Programming, we are ready to explore and do a few more exercises to think logic, use low end computing tools such as Pointers, and implement it in C programs. We will also get an opportunity to implement Strings and explore a new territory of Files in C programming that was not directly discussed in class.

At any duration during and/or after the lab, students are recommended to team up with the Professor and the TL(s) to clarify if there is any confusion related to the lab and/or class materials. The Professor proofread the document more than once, if there is an error in the document, it will be much appreciated if you can communicate that to the Professor. The class will be then informed as soon as possible regarding the error in the document. Additionally, it is highly recommended that students will reach out to the Professor in advance of the lab submission with any questions. Waiting till the last minute will minimize the student’s chances to get proper assistance from the Professor and the Technical Leader(s).

Students are recommended to get started with this part in the laboratory session, by discussing ideas and clarifying with the Professor and the Technical Leader(s). It is acceptable to discuss high-level ideas with your peers, while all the work should be done individually. Late submission is accepted for the part(s) in this section, based on the late policy outlined in the course syllabus.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.

2. Submitting a copy of the other's program(s) and technical reports is strictly not allowed. Please note that all work done during lab sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work individually, students maximize the learning and increase the chances to do well in other assessments such as Quizzes, exams, etc · · ·

Preliminary Steps

It is important that you can set up Docker and GitHub to complete the rest of the lab. Please follow the guidelines below to complete the preliminary steps.

1. **[Docker Setup.]** At this point, I expect the MAC, Linux, and Windows Pro users, to have this step completed based on our previous class discussions. For those who had not completed this step, the documentation below should provide more details regarding the download and installation setup.

   - Get Docker setup completed on your laptops:
   - Docker Mac Setup:

     `https:/docs.docker.com/docker-for-mac/install/`
   - Docker Ubuntu Setup

     `https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04`
   - Docker Windows Setup:

     `https:/docs.docker.com/docker-for-windows/install/`
   - If the setup goes correctly as desired, you should be able to get started and validate the Docker version and run the hello world docker container using the following commands:

     docker –version

     docker run hello-world
   - There are some more documentation for Docker get started to test your installation in the link provided below:

     `https:/docs.docker.com/docker-for-mac/`

     `https:/docs.docker.com/docker-for-windows/`

2. **[Loading Docker Container.]** There are two steps in loading the container, namely:

   - Build the container
   - Connect and Run the container

   Build the container: So to build the container. the following steps should be performed.

   (a) First, accept the lab URL provided in Slack. After downloading the lab folder from the GitHub classroom, navigate to the cmpsc200-fall-21-lab02 directory using terminal (Mac/Ubuntu) or Command Prompt/Docker quick start terminal (windows).

   (b) Build the docker image using the following command:

   docker build -t cs200lab02 .

   Please note, you are required to have the period in the command above.

   (c) Note: In the command above, cs200lab02 is the user-provided image name. This could be random. But it is recommended to use the same name to easily follow the rest of this document. Additionally, it is required to be inside the `cmpsc200-fall-21-lab02` directory to run the build command. If you are not inside the `cmpsc200-fall-21-lab02` directory, you may receive an error message.

   (d) Upon successful build, it is recommended to verify the correctness of image creation by using the following command:

   docker image ls

   (e) The image named ”cs200lab02” should be listed as one of the outputs from the command above.

   Connect and Run the container: So to create and run the container. the following steps should be performed.

   (a) Run the docker container based on the image created in the previous steps using the following command:

   Mac/Ubuntu:

   ```
   docker run --rm -v $(pwd)/src:/root -it cs200lab02
   ```

   Windows:

   ```
   docker run --rm -v "%cd%/src":/root -it cs200lab02
   ```

   (b) To run the above command, it is required to be inside the `cmpsc200-fall-21-lab02` directory. And, please note, you will log in to the container after entering the above command.

   (c) After creating the container, the run command above creates a mount between the host machine and the container with a shared folder space. So, any files placed inside the host mount directory can be easily accessible inside the container mount directory and vice versa.

   (d) After connecting to the container, we can compile C Programs using the command below within the container:

   gcc hello.c -o hello.out

   (e) After compiler the program, we can execute C Programs using the command within the container:

   ./hello.out

   (f) **[GitHub Setup.]** Take a look at the detailed documentation for getting started with GitHub, which is available at: `https://docs.github.com/en/get-started`

   You are required to know the procedure to git clone, git pull, git add, git commit, and git push to access the lab specification folder and to submit your lab for grading purposes. If there is an issue with your GitHub setup please discuss it with your Technical Leader(s) and/or the Professor.

Section 1: Basket Ball Players Program

This section is worth 15 points. The points breakdown is provided below:

- Task 1 = 20 points

We implemented a nested for-loop program, to iterate through an array both by using indexing through square brackets and by using pointer arithmetic. The starter code is provided with a series of Basket ball player names and their rankings. Our goal is to form two teams of 5 players each. The players provided in the names and rankings array are arranged in the following order:

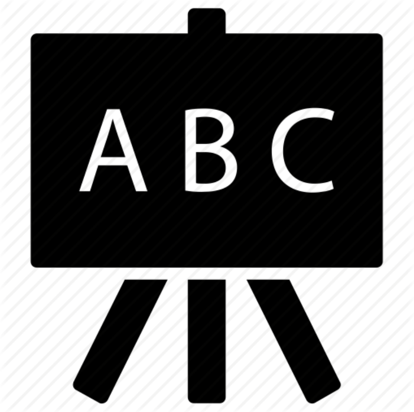
1. Even number indices is classified as **Team-A** candidates.
2. Odd number indices is classified as **Team-B** candidates

Your program is expected to create the groups by including the highest ranking candidates in both teams. **Important** to note that you can only include the candidates listed in Team-A to form Team-A. Similarly, you can only include the candidates listed in Team-B to form Team-B. The **teamA** method accepts three inputs, that is the names array, rankings array, and size of the arrays. This method is expected to show a report of all the 5 Team-A players after performing the required logic. The **teamA** method accepts three inputs, names array, rankings array, and size of the arrays. as well . This method is expected to display a report of all the 5 Team-B players after performing the required logic.

Key point to note: You are required to prepare the data internally in such a way that you are able to first get the Team A and B candidates separetely. At that point, you may start doing the next step of processing. The next step is to set up the iterative block of code to identify the key players based on their rankings and then display the team details on the console. It is not allowed to edit the main program and/or display the result by a manipulating the array manually. During grading, the input array values will be changed and the systematic/logical solution will be tested and evaluated. The **output** displayed should include the names of the 5 players and their ranking in the teams, corresponding to the team selected by the user through the user prompt.

Although you may use any number of arrays and variables (as an intermediate) layer of storage to perform this requirement, it is very important to understand that the best logical solution is to use the least number of intermediate arrays and variables. The least arrays and variables we use, the more effficient the code is. To use the least arrays and variables, we need to think about the logic more deeply. Note: this task requires some thinking and the goal of this course to make you think deeply as we navigate the intricacies of low end computing together. Please don't start coding immediately. Think logic on paper first and talk to the Professor and the TL about the design (logic) then start coding. To this end, the Professor would like to share a note that: **Best programs are developed in paper first and then implemented on computer.**

Section 2: Strings Program.



This section is worth 20 points. The points breakdown is provided below:

- Task 2 = 10 points

Write a simple functionality to detect if the given word is a Palindrome using the starter code file named called Palindrome.c that is expected to performs the following functionality:

1. Prompt the user to type in an input string and store the string into a char pointer array of predefine size 20. That is, we only check the words of length 20 or less.
2. Implement the logic in the palindrome method to validate if they are equal. If they are equal then print ”The word is a Palindrome”, else print ”The word is not a Palindrome”. For example ”DAD”, ”MOM”, ”CIVIC”, ”KAYAK”, etc.. are palindromes.
3. You are only allowed to use character arrays to implement this part. The usage of any external string libraries, implementation using strcmp, pointers, and other library based functions is not acceptable.
4. It is strongly recommended to use Pointer arithmetic to iterate the array. However, I will not stop you from using the indexing method using square brackets if you prefer to do so.

Section 3: Experiencing an unexplored territory of C

This section is worth 20 points. The points breakdown is provided below:

- Task 3 = 10 points
- Task 4 = 10 points

It is worth making a note, that the file system is just the output component referenced in the Von Neumann architecture. So far, we primarily used the Display monitor screen, as the output component to process our programs. In this section, we will experience an unexplored territory of developing C programs using the File System. To complete this part, it is required to have a solid understanding of the basic concepts of C Programming from class discussions and the previous two labs. It is important to review the slides, reason through and understand the logical tasks (binary using modulo 2) discussed, and also complete the reading assignments as required. There is a C program provided in the starter-code provided repository, to help stay focused on the implementation details, and to get started with the development of this part. To complete this part, it is required to do the following:

- On page 17 of K&R there is a program to copy a file. To run it on a file named “myfile.txt”, you type:

  ```
  ./a.out < myfile.txt
  ```

  Review the starter-code in a file named `parser.c` and `ritchie.txt`. The code is modified in the `parser` program so that it prints out only every other character (other than the newline character, which will always be printed), i.e., the first, third, fifth, seventh,· · · This implementation is done by adding a counter and only printing when the counter has an even value. The text file named ”ritchie.txt” can be used for test data, which is also available in the ”src” directory. Read the comments to make sure you understand how the program works and the prompts to get the user input. This understanding is important to complete the tasks outlined below:

  1. **Task 3:** Write a C program modeled after the ones in sections 1.5.3 and 1.5.4 of K&R that do the following: Given an input file, print it with leading line numbers, starting with line 1. Assume there are no more than 999 lines in the file. Line numbers should be right-justified in the first 3 columns. Submit your work, using a new file named `file1.c`.
  2. **Task 4:** Write a C program modeled after the ones in sections 1.5.3 and 1.5.4 that does the following: Given an input file, count the number of vowels and consonants and print out these counts, appropriately labeled. The vowels are a, e, i, o, and u (both upper and lower case); all other letters are consonants. You may not use any built-in C functions for checking for upper-case, etc. Submit your work, using a new file named `file2.c`.

     The text file named ”ritchie.txt” can be used for test data, which is also available in the ”src” directory.

Section 04 - Honor Code

Make sure to **Sign** the following statement in the `honor-code.txt` file in your repository. To sign your name, simply replace Student Name with your name. The lab work will not be graded unless the honor code file is signed by you.

This work is mine unless otherwise cited - Student Name

Section 05 - Reflection

Add a Reflection to the repository by modifying the `reflection` file in the lab repository. List out the biggest learning points and any challenges that you have encountered during this lab.

Submission Details

For this assignment, please submit the following to your GitHub lab repository.

1. updated version of **basketball.c** file.
2. updated version of **palindrome.c** file.
3. updated version of **file1.c** file
4. updated version of **file2.c** file
5. A document containing the reflection of the lab in the file named `Reflection`.
6. A signed honor code file, named `Honorcode`.
7. To reiterate, it is highly important, for you to meet the honor code standards provided by the college. The honor code policy can be accessed through the course syllabus.

Grading Rubric

1. There will be full points awarded for the lab if all the requirements in the lab specification are correctly implemented. Partial credits may be awarded if deemed appropriate.
2. Failure to upload the lab assignment code to your GitHub repository will lead to receiving no points given for the lab submission. In this case, there is no solid base to grade the work.
3. There will be no partial credit awarded if your code doesn't compile correctly. It is highly recommended to validate if the correct version of the code is being submitted before the due date and make sure to follow the honor code policy described in the syllabus. If it is a late submission, then it is the student's responsibility to let the professor know about it after the final submission in GitHub. In this way, an updated version of the student's submission will be used for grading. If the student did not communicate about the late submission, then automatically, the most updated version before the submission deadline will be used for grading purposes. If the student had not submitted any code, then, in this case, there are no points awarded to the student.
4. If a student needs any clarification on their lab grade, it is strongly recommended to talk to the Professor. The lab grade may be changed if deemed appropriate.

Wish you
all the best