

CS 200 Midterm Exam

CS 200
Fall 2020
Midterm Exam
10/14/2020
Time Limit: 1:50 hours

- **Exam Structure:** This exam contains two parts. Part 01 is the theoretical exam. Part 02 is a practical exam. The theoretical exam part contains 10 questions (a mix of multiple-choice and descriptive questions) provided through an online portal. The practical exam part contains one or more programming requirements outlined in this document that require implementing code files to address all the requirements provided.
- **How to take this exam?** You may *not* use your books, notes, phone, and other resources on this exam. You are allowed to use a resource sheet (1 page can be 2 sided) handwritten or typed notes in order to help you to do well in this exam.

Think carefully before answering the following questions. After the answer submitted, then you may not be able to change it!

To prioritize health and to follow the college's guidelines on physical distancing, this exam is administered online. Please note: All test takers are required to sign a pledge to abide by the class honor code. Any violation(s) will be considered fraud, and outside the norms of staying true citizens of the gator community. In-person attendance is not required during the exam day. The skill test has a time limit set for 1 hour and 50 minutes. The test will begin once a student begins the exam and does not pause if the student navigates away from the test. It is required for everyone to take the test during the scheduled time. The link to the exam will be in open status during the lab timings, which is from 3:00 PM to 4:50 PM. If extra time is needed, then the student should contact the professor as soon as possible, and an arrangement will be done to accommodate the request.

- **Additional Details:** I had tried to proofread the exam carefully, but if you see something that looks like an error, please notify me immediately so that I can look at it and possibly correct it. I had tried my best to create the questions so that it is simple to understand and easy to read, but if you find the choice of words to be confusing or unclear, then you should take a break and stop by at my Slack space as soon as possible so that I can clarify.
-

EXAM SHEET

1. (50 points) **PART 1:** Navigate to the online portal link below and follow the **Part 1 guidelines** provided in the portal.

<https://forms.gle/9ERCGs5N5ruFHmgU8>

2. (50 points) **PART 2:** Provide detailed implementation for the requirements outlined below:
Hard Requirements: There will be full points awarded if you implement all the requirements thoroughly as indicated. Partial points will be awarded if there is any compromise done to implement any of the requirements listed below. If the code does not compile and/or throw any runtime errors, then there will be additional points taken. So be cautious to make necessary backups during the exam when you have working code and check the time regularly to make sure you finish the requirements within the allocated period. Add necessary comments in your program files, to maximize your chances to receive partial credit.

Make sure to add the following text as a comment in all your code files, including **binary.c** and **arrays.c** in the repository shared. Enter your name at the end of the honor code statement.

This code is written following the CS-200 honor code policy and exam policy outlined in the document above and all the code files submitted were completely implemented by me.

Please note, as per the college policy, we are unable to **grade** your work if you fail to add the statement above in all your code files. All your commits are logged in GitHub so please make sure to follow the protocols outlined above.

1. **Binary Match Table [25 points]:** The **binary.c** program, prompts the user to input a number greater than 0. The starter code provides a platform to implement and print the binary equivalent of the user-defined decimal number in both forward and backward directions. The forward method in the started code is fully implemented. This method prints the binary match table in the forward/ascending direction. To complete this part of the exam, it is required to do the following:
 - Carefully read through the code file, and identify how the starter code works. It may be required to compile and execute the program a few times to fully grasp and understand the logical flow of the starter code.
 - Implement the backward method, to print the binary match table in the backward/descending order.
 - Once, the backward method is implemented, the code should be tested out. The expected result should be in the opposite order of the result printed from the results printed in the forward direction.
2. **Array Processor [25 points]:** The **arrays.c** program, prompts the user to input two numbers, for row and column sizes. The starter code provides a platform to load and print the values in the two-dimensional array space (dynamically) using the malloc feature. There are two schemes used in the starter code program. The first scheme (scheme1 method), is fully implemented. This scheme loads the array incrementally starting from 1. To complete this part of the exam, it is required to do the following:
 - Carefully read through the code file, and identify how the starter code works. It may be required to compile and execute the program a few times to fully grasp and understand the logical flow of the starter code.
 - Implement the scheme2 method, to load and print the dynamic malloc based two-dimensional array, with contents incremented in the power of the numbers from

scheme1 for each slot. The numbers in this scheme are 1, 4, 9, 16, 25, 36, ... A sample screenshot of a 4 by 4 array using scheme2 is provided below for reference:

```
amohan@ALDENV8075 Downloads % gcc arrays.c -o arrays.out
amohan@ALDENV8075 Downloads % ./arrays.out
Please enter the number of rows, (0) for exit:4
Please enter the number of cols:4
Please enter the scheme number:2
1      4      9      16
25     36     49     64
81     100    121    144
169    196    225    256
Please enter the number of rows, (0) for exit:0
```

- Once, the scheme2 method is implemented, the code should be tested out. The expected result should be matching the data points from the sample output above. Of course, the output above is for a 4 by 4. The program will be run for different row and column sizes.

Self Reflection: It is required that we complete a self-reflection about your exam preparation, exam performance, and provide feedback about the exam and our CS-200 class. The reflection document may be added as a markdown file or as a PDF file named **reflection**. Add a self-reflection document with the following details:

1. What did you do from your side to do well in this exam? (Feel free to comment on anything).
2. What did you think of this part of the exam? Feel free to comment on anything — fairness, length, coverage, difficulty, or anything else.) Any constructive criticism is welcome.
3. What are the most helpful aspects that you like us to continue doing in our classes?
4. What would you like us to start doing more to further enhance your learning in our classes?