Assignment 1 (5% of Course Total)

Due date: 11:59pm, Sep 27, 2024

This is an individual assignment. Part of the assignment will be graded automatically. Make sure that your code compiles without warnings/errors and produces the required output. Also use the file names and structures indicated as requested. Deviation from that might result in 0 mark.

Your code MUST compile and run in the CSIL machines using the steps covered in class (see the **Before You Submit** section for details). It is possible that even with warnings your code would compile. But this still indicates there is something wrong with you code and you have to fix them, or marks will be deducted.

Your code MUST be readable and have reasonable documentation (comments) explaining what it does. Use your own judgement, for example, no need to explain i += 2 is increasing i by 2, but explain how variables are used to achieve something, what a certain loop is doing, and the purpose of each #include.

Description

There is a total of 3 questions in this assignment. For each question, write your answer in a single file along with the required information stated in the question. **Unless otherwise specified, do not include any pre-existing libraries in your answers**. You can however write your own helper functions. Also, do not print anything unless the question asks you to. **None of these files should contain the main function**.

Question 1 [5 marks]

Write a function that replaces a specific digit in an int number with another digit and returns the result as an int. Return the same number if the target/replacement character is not a digit. Use this function header:

int replaceDigits(int number, char target, char replacement)

For example:

replaceDigits(1, '1', '2') should return 2
replaceDigits(-232, '3', '0') should return -202
replaceDigits(123, '4', '5') should return 123
replaceDigits(1223, '!', '?') should return 1223
replaceDigits(5670, '5', '0') should return 670 (leading zero will not be part of the resulting number)
replaceDigits(30400, '0', '9') should return 39499

You can assume the number does not have leading zeros (e.g., we will not call *replaceDigits(01, '1', '2')*), except when the number is actually zero (i.e., we might call *replaceDigits(0, '1', '1')*).

Only include the a1_question.h header file and the function definition (and your helper functions, if any) in the source file and name it as **a1_question1.c**. Do not use recursion in your answer.

Question 2 [5 marks]

Write a function that takes in 4 parameters: an int array, its size, the left index, and the right index; and checks if the elements between the left and right index (inclusive) in the int array are sorted in ascending order. If the left index is larger than the right index, swap them first. Then, if the left index is invalid (e.g., negative, larger than the size), use the leftmost valid index of the array; if the right index is invalid, use the rightmost valid index of the array. Use this function header:

bool rangedCheckforSorted(int array[], unsigned int size, int leftIndex, int rightIndex)

For example (suppose there is a myIntArray created as [-4, 3, -12, 0, 5, 72, 88, 128, 1, 64]):

rangedCheckforSorted(myIntArray, 10, -2, 3) will check indexes between 0 and 3, and return false
rangedCheckforSorted(myIntArray, 10, 7, 2) will check indexes between 2 and 7, and return true
rangedCheckforSorted(myIntArray, 10, 5, 5) will check indexes between 5 and 5, and return true
rangedCheckforSorted(myIntArray, 10, 0, 9) will check indexes between 0 and 9, and return false

You can assume the array has one or more elements and the size is always correct. Content of the int array does not change after calling this function.

Only include the a1_question2.h header file, <stdbool.h>, and the function definition (and your helper functions, if any) in the source file and name it as **a1_question2.c**. Do not use recursion in your answer.

Question 3 [6 marks]

In C arrays can be accessed using pointers. As an exercise, write a function that reverses the polarity of every element in a 2D int array (i.e., -1 becomes 1, 46 becomes -46, …etc.) **using only pointers**. That is, your code must not use any [ ], which is the non-pointer version of array access. Use this function header:

*void reversePolarity(unsigned int row, unsigned int col, int** array)*

For example, suppose we have a 2-by-3 array my2DIntArray with the content [[0, 1, -2], [-999, 2345, 678]]. Then the function will be called as (the array needs to be created differently, see the test file):

reversePolarity(2, 3, my2DIntArray)

And when the function returns, the content of my2DIntArray becomes [[0, -1, 2], [999, -2345, -678]].

You can assume that the row & col parameters always correctly indicate the number of rows & columns of the array. **Do not use [ ] anywhere in your answer** (if you do you get 0 for this question).

Only include the a1_question3.h header file and the function definition (and your helper functions, if any) in the source file and name it as **a1_question3.c**.

Coding Style [4 marks]

Your program should be properly indented, have clear and meaningful variable names (e.g., no single-letter variable names except loop iterators) and enough white space and comments to make it easy to read. Named constants should be used where appropriate. Each line of code should not exceed 80 characters. White space should be used in a consistent manner. Remember to include your information.

To help you to get into the habit of good coding style, we will read your code and marks will be deducted if your code is not styled properly.

Before You Submit

Before submitting your assignment, make sure you have test-run your code in our CSIL machines using the steps covered in class: **open your assignment folder in VS Code, use the gcc command with options in the terminal to compile your code, and run the executable**. We will use these steps when marking your submission and will deduct marks if your code does not compile/run – to maintain fairness we do not accept reasons like “but it worked on my computer” or “but it ran just fine in my IDE”.

The Makefile provided in this assignment is used by a command in the CSIL machines called “make” to quickly compile your code. It is especially useful if you have multiple source files. To use it, type the following command in the prompt (make sure you are in the directory with all the files of Assignment 1):

```
$ make test1
```

The example above illustrates how Question 1 is compiled into an executable called “test1” when using the Makefile. Replace the “test1” with “test2”, “test3”, …etc. for other questions. You can then run the executable by typing **./test1** to test your code for Question 1. If you make changes to your code, use the make command again. You can also use **make all** if you want to compile all your code at once.

The test files (test1.c, test2.c, …etc.) are provided in this assignment for you to test your code. Each typically contains a main function along with other tester functions and/or calls. You can modify them to further test your code, but do not submit these test files because we will be using our test files that are similar but not identical to grade your assignment. This makes sure that your code is not written to produce hard-coded output.

The header files (question1.h, question2.h, …etc.) are there to make the compilation work. You can look at them but do not modify them. You also do not have to submit them.

Submission

Submit **only the 3 source files** indicated above (a1_question1.c, a1_question2.c, a1_question3.c) to CourSys. Refer to the corresponding Canvas assignment entry for details.

Assignment late penalty: 10% per calendar day (each 0 to 24 hour period past due), max 2 days late.

Academic Honesty

It is expected that within this course, the highest standards of academic integrity will be maintained, in keeping with SFU’s Policy S10.01, “Code of Academic Integrity and Good Conduct.” In this class, collaboration is encouraged for in-class exercises and the team components of the assignments, as well as task preparation for group discussions. However, individual work should be completed by the person who submits it. Any work that is independent work of the submitter should be clearly cited to make its source clear. All referenced work in reports and presentations must be appropriately cited, to include websites, as well as figures and graphs in presentations. If there are any questions whatsoever, feel free to contact the course instructor about any possible grey areas.

Some examples of unacceptable behavior:

- Handing in assignments/exercises that are not 100% your own work (in design, implementation, wording, etc.), without a clear/visible citation of the source.
- Using another student's work as a template or reference for completing your own work.
- Sharing your work with or making your work available on any platform for anyone who would benefit from it (e.g., other students in the course).
- Using any unpermitted resources during an exam.
- Looking at, or attempting to look at, another student's answer during an exam.
- Submitting work that has been submitted before, for any course at any institution.

All instances of academic dishonesty will be dealt with severely and according to SFU policy. This means that Student Services will be notified, and they will record the dishonesty in the student's file. Students are strongly encouraged to review SFU's Code of Academic Integrity and Good Conduct (S10.01) available online at: http://www.sfu.ca/policies/gazette/student/s10-01.html.

Use of ChatGPT or Other AI Tools

As mentioned in the class, we are aware of them. I see them as helpers/tutors from which you can look for inspiration. However, these tools are not reliable and they tend to be overly confident about their answers, sometimes even incorrect. It is also very easy to grow a reliance to them and put yourself at risk of not actually learning anything and even committing academic dishonesty. Other issues include:

- When it comes to uncertainty, you won't know how to determine what is correct.
- If you need to modify or fine tune your answer, you won't know what to do.
- You will not be able to learn the materials and thus will not ne able to apply what you learn in situations where no external help is available, for example, during exams and interviews.

For introductory level courses and less sophisticated questions, it is likely that you'll get an almost perfect answer from these tools. But keep in mind if you can get the answer, everyone can also get the answer. Bottomline is, if you ask these tools to give you an answer and you use the answer as yours, you are committing academic dishonesty by claiming work that is not done by you as yours. You can, however, ask general questions like "how do I write a for-loop?", "explain mergesort to me" and use the answers to help you come up with your own answers – make sure you have cited it at the top of your code/essay as comments by stating what tool you used what questions you asked, and how you used the answers. Type everything yourself.

Note that different instructors might have different policies regarding the use of these tools. Check with them before you proceed with assignments from other courses.