

MECHTRON 2MP3 : Midterm Test

Fall 2022

Nicholas Moore

Maximum Grade: 24/24

Special Instructions - Please read

YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE. BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

Permitted Materials:

(cell phones are not permitted)

- Laptop Computer
- McMaster Standard Calculator (CASIO FX-991MS/MS+)
- One Double Sided 8.5 x 11 Crib Sheet

Materials To Be Supplied To Students:

- Scrap Paper

Special Instructions - Please Read

- This exam contains 3 questions on 3 pages.
- Do not rename the files contained in the “Exam.zip” compressed archive.
- Do not use function names other than those provided.
- Do not change the arguments of functions. Use them as provided or indicated in the question.
- You must submit your exam using your CS1XC3 private github repository. Your final commit must be timestamped prior to the end time of the exam.
- There is no grace period for exam submissions.
- Code submitted with syntax errors is worth zero. Commands which don't execute are worth zero. no exceptions.

Submission Instructions

- Create a new folder called **T** in your **MT2MP3** submission repository on the department's gitlab server.
- Code templates are available as a zip file from Avenue. You may simply decompress the archive inside **MT2MP3/T**, and the correct directory structure and file names will be added automatically.
- Be sure to **add**, **commit** and **push** your answers *before* 9:00am EST, Sunday November 7th. You can verify your submission via web browser by inspecting the contents of your submission repo.

Remember! Failure to commit is commitment to failure!

Good luck!

Questions

1. (8 points) The Recamán Sequence

The Recamán Sequence is a mathematical sequence which is kind of spooky to listen to if you interpret the numbers as musical notes. It works like this:

- No number appears in the sequence more than once.
- The sequence is not allowed to contain negative values.
- The sequence starts at zero.
- To calculate the n^{th} number in the sequence:
 - We consider the value of the $(n - 1)^{th}$ number.
 - If this value **subtract** n does not exist in the sequence so far, it becomes the next number.
 - Otherwise, the next number is this value **plus** n .

It goes like this:

n	0	1	2	3	4	5	6	7	8	9	10	11
	0	0 + 1	1 + 2	3 + 3	6 - 4	2 + 5	7 + 6	13 + 7	20 - 8	12 + 9	21 - 10	11 + 11
$R(n)$	0	1	3	6	2	7	13	20	12	21	11	22

Write a function, `getRekt`, with the following arguments:

- `int n` \Rightarrow the size of the Recamán sequence we are calculating.
- `int* numAdditions` \Rightarrow implicit output value, the number of times you added in the sequence as calculated.
- `int* numSubtractions` \Rightarrow implicit output value, the number of times you subtracted in the sequence as calculated.

Your function must output the n^{th} number in the sequence. You may assume that $n \geq 0$ and that both integer pointer arguments will point to valid memory.

2. (6 points) High Fives

It's a little known fact that the CAS department has an policy of obligatory high-fives in the halls of ITB. Whenever two students pass in the hallway, a high five must occur.

We will encode hallways as a string, with the `>` character indicating a person travelling right, and the `<` character indicating a person travelling left. The `-` character indicates a section of hallway with no students in it.

Write a function in C named `highfives` which takes a character array, and outputs the number of high fives that must occur between all of the students in the section of hallway. For example, the following hallway would have 3 high fives occur:

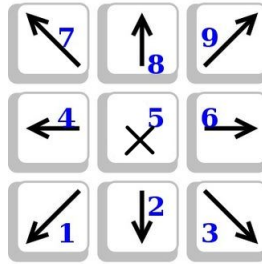
- `>>>-<`

Additional, more complicated examples are provided in your code template.

3. (10 points) A Random Walk

Professor Confunkulus enjoys taking refreshing strolls in the evening, but because he is a mathematics professor, he wishes to make his evening walks as mathematically convoluted as possible.

He takes a field, and divides it into a square matrix of size 10. to each cell in the matrix, he assigns a random number between 1 and 9. Each number represents a direction, as one would see on a typical keyboard number pad:



Professor Confunkulus moves through the field one cell at a time, always following the direction indicated in the cell he's currently standing in. He stops if:

- He is currently standing in a 5-cell.
- His walk would take him outside the field.
- His walk would have him return to a previously visited cell.

Your mission, should you chose to accept it, is to write a C function which takes a 10x10 array of integers and a pointer to a starting location within that array, and outputs a pointer to his final position within the array.