University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

AUTUMN TRIMESTER EXAMINATION 2021/2022

ACM40660

Scientific Programming Concepts

Prof Adrian Ottewill

Dr Barry Wardell

Mr Christopher Werner and Dr Buket Benek Gursoy

Time Allowed: 2:00 hours

Instructions

- The are two coding challenges. Download the code stubs from BrightSpace and complete the codes.
- The exam must be completed individually.
- You may use the following materials:
  1. Lecture notes, assignments and all materials on the Brightspace page for the module.
  2. Any other hand-written notes that you have prepared.
  3. Any textbooks, in either print or electronic form.
- With the exception of the above, **no other sources are allowed**. In particular, **you may not use web search or other web pages**. You also **must not discuss the exam with anyone else**.
- To confirm that you have complied with the School of Mathematics and Statistics Honour Code, please sign and submit the file titled **'SMS Exam Honour Code'**, underneath "I have neither given, sought, nor received aid during this examination."
- Once the exam has finished, upload your final codes to Brightspace. We recommend that you run and compile your code using sciprog.ichec.ie. If you still have issues with getting data onto and off sciprog, check out the videos in the Information tab.
- **An additional 30 minutes** has been provided after the exam finishes for scanning and uploading. Any submissions (other than those arising from technical difficulties or other extenuating circumstances) after this time will be penalised by multiplying the mark by a factor that decreases linearly from 100% to 0% over 30 minutes.
- If you encounter technical difficulties that lead to a delay, please continue to upload your solutions as soon as possible. Brightspace will accept late submissions, but will flag them as late. After you have uploaded your solutions and if the allowed 30 minutes have passed, please email me at **christopher.werner@ichec.ie** to explain the reason for the delay. Please also include a photograph or screenshot of your upload attempt and of any error messages you have encountered.
- If you have any queries or difficulties during the exam, we will be available in the Zoom meeting link below. $https://nuigalway-ie.zoom.us/j/95469716649?pwd=Z0NSUWpvdHRiNVNVbEorYXhLc2IwQT09$
- I will also actively monitor my email as an alternative in case you have difficulties using Zoom.

Questions

1. Laurent series for a complex function $f(z)$ about a point z_0 is given by

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n + \sum_{n=1}^{\infty} \frac{b_n}{(z - z_0)^n}$$

Given $f(z) = \frac{\exp(z)}{z^2}$, use the following Laurent expansion of $f(z)$ around $z_0 = 0$ to calculate an estimated value by setting $z = 1.57$.

$$f(z) = \frac{1}{2!} + \frac{z}{3!} + \frac{z^2}{4!} + \frac{z^3}{5!} + ... + \frac{1}{z} + \frac{1}{z^2}$$

This program will read in an array of elements from a file. This array represents a set of parameters a_i in the series above. The value of the parameters in the second sum is $b_1 = 1, b_2 = 1$ and $b_i = 0$ for all $i = 3, 4, ...$

(a) Use the program `laurent_stub.c` or `.f90` as the starting point.

(b) Get the stub and the file `laurent.csv`, from BrightSpace.

(c) The code to determine the length of the parameters vector corresponding to a_i is already given. Complete the code as explained in TODOs.

(d) Create a dynamic array using `malloc` to store the vector elements and read the file.

(e) Calculate the approximated value of $f(z)$ and compare the estimate against the exact value.

2. We are going to implement a binary tree with a single integer as its data. A binary tree is a tree in which every node has at most two children, which are referred to as the left child and the right child. A node's left child must have a value less than its parent's value and the node's right child must have a value greater than its parent's value. The tree will be manipulated with several functions. These functions will often take pointers as arguments, and some of them will return pointers as a return value. Test your functions from your `main()`. A visual reference is provided further down for an illustration.

(a) Use the file `binarytree_stub.c` or `.f90` as the starting point for your code. The nodes of the tree have already been created with `struct node`. Moreover, the following function has been written

- The `newnode()` function that allocates a new node in memory and fills its content with a value passed as an argument.

Other functions needed have been partially done. Complete the code as explained in TODOs.

(b) Write the `insert()` function that inserts a new node to tree. Insert the new node to the left subtree if its content is less than that of the root or the right subtree if its content is greater than or equal to the root. The function should be recursive.

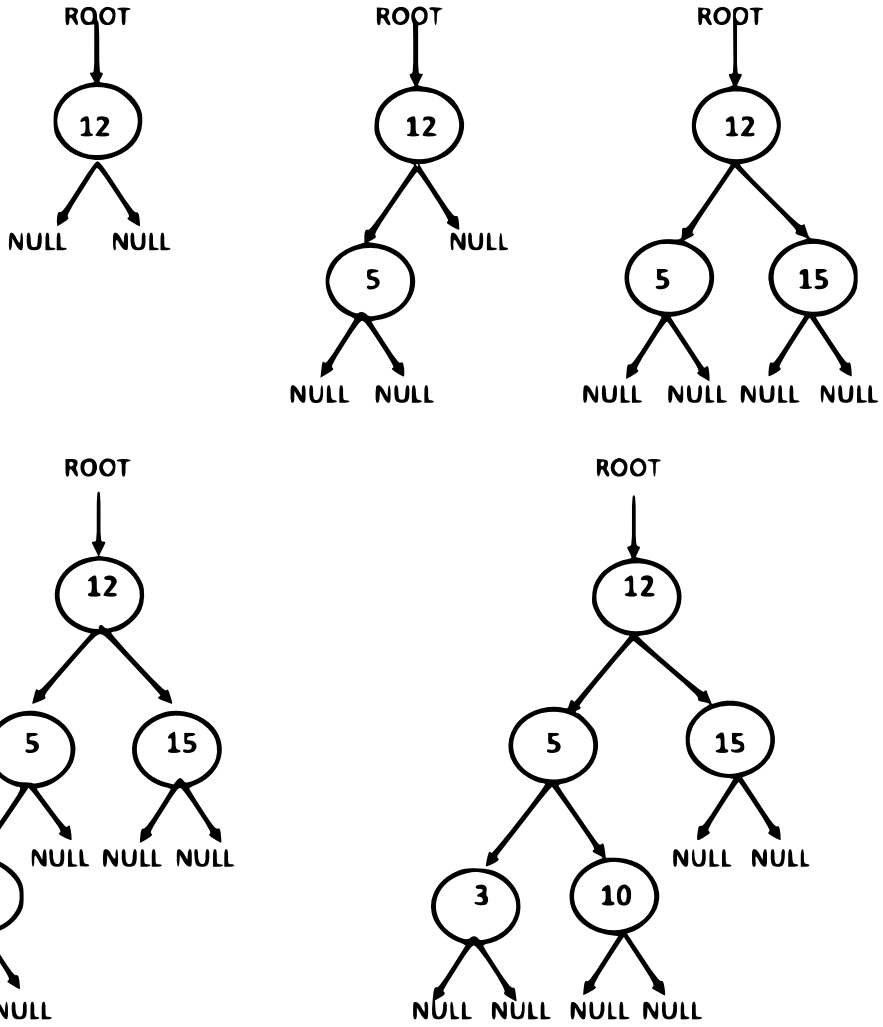
(c) Write the `printinorder()` function that prints the content of the nodes of a tree, whose root node passed as an argument, in the order of left node, root node, and then right node.

(d) Write the `sumnodes()` function which iteratively or recursively calculates the sum of all contents in a tree and returns that sum.

(e) Write the `findminimum()` function which calculates the minimum value of all contents in the tree and returns the node which has minimum value in C and returns the minimum value in Fortran. The function should be recursive.

(f) Write the `search()` function that searches a given content in the tree and returns the searched node in C. It returns TRUE (if found) or FALSE (not found) in Fortran. The function should be iterative.

(g) Write the `deletetree()` function that deletes a given tree and prints the content of deleted nodes. It returns the root node in C. The function should be recursive.



Some general points:

(a) the main point is to get the program to do what the question asks (use either FORTRAN or C not C++),

(b) make sure your code is syntactically correct and standard (i.e. it compiles on sciprog),

(c) comment your code and place any observations as comments,

(d) make sure you upload the completed code to BrightSpace.

(e) good luck!

—o0o—