UNIVERSITY OF GREENWICH

COMP1819 **Algorithms and Data Structures**	2023/24	**Coursework ID:**	**Contribution: 50% of course**
Coordinator: **Dr Tuan Vuong**	**Release Date:** **05/2024**	**Deadline Date:** **12/07/2024**	**Return Date:** **02/08/2024**
This coursework should take an average student who is up-to-date with tutorial work for approximately 25 hours. Feedback and grades are normally available within 15 working days of the coursework deadline.			
Learning Outcomes: 1. Select and employ data structures appropriate to various problems. 2. Understand the relationship between algorithms and data structures. 3. Formulate and solve programming problems using learned concepts. 4. Understand the complexity of algorithms in terms of time and memory.			

Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing coursework from another student and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University. Please see your student handbook for further details of what is / isn't plagiarism.

All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using. Code snippets from open-source resources or YouTube must be acknowledged appropriately.

Your work will be submitted for plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.

Coursework Submission Requirements

- An electronic copy of your work for this coursework must be fully uploaded on the Deadline of **Friday 12/07/2024** using the link on the coursework Moodle page for COMP1819.
- For this coursework you must submit a single PDF document. In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand-written mathematical notation, but when scanning do ensure the file size is not excessive.
- For this coursework you must also upload a single **ZIP** file containing supporting evidence.
- There are limits on the file size (see the relevant course Moodle page).
- Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.
- Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.
- You must NOT submit a paper copy of this coursework.
- All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See http://www2.gre.ac.uk/current-students/regs

Algorithms and Data Structures (ADS) - COMP1819

Develop and optimise solutions in Python with ADS and provide complexity analysis.

This is an INDIVIDUAL coursework.

Detailed Specification

As part of a summer internship, you are joining a growing start-up company in the cryptography domain. You are asked to solve a coding interview question in Python that tests a candidate's knowledge and skill in both basic data structures and algorithms with different levels of optimisation.

The question:

Create a Python program that helps you find special numbers! You'll input two positive numbers, m and n (where m is smaller than n), and the program should tell you how many special numbers are between them (inclusively). The program should not only count these special numbers but also show them to you.

If there are less than 6 special numbers, it should display all of them. Otherwise, it should show you the first three smallest special numbers and the last three biggest special numbers.

Special numbers are those that are both prime (only divisible by 1 and themselves) and palindromic (read the same backwards as forward).

The program must not use a hard-coded list of values (you can utmost have 5 values hard-coded) or an external data file to assist the computation. The program is encouraged to finish within 1 hour (run-time) for any test case. Each solution should be your own work, and it's allowed to use Python's built-in libraries. Try to keep your program concise, not exceeding 100 lines.

Examples:

Input (m n)	Output (Total: Special Numbers)	Comments
1 3	2: 2, 3	List of special numbers = 2,3 Total number of special numbers = 2
1 20	5: 2, 3, 5, 7, 11	List of special numbers = 2, 3, 5, 7, 11 Total number of special numbers = 5
2 200	10: 2, 3, 5, 151, 181, 191	Total number of special numbers = 2, 3, 5, 151, 181, 191 Total number of special numbers = 10

Test cases:

#	Input (m n) (extra space for readability)	Output (Total: Special Numbers)
1	1 2_000	Your answers required
2	100 10_000	Your answers required
3	20_000 80_000	Your answers required
4	100_000 2_000_000	Your answers required
5	2_000_000 9_000_000	Your answers required
6	10_000_000 100_000_000	Your answers required
7	100_000_000 400_000_000	Your answers required
8	1_100_000_000 15_000_000_000	Your answers required
9	15_000_000_000 100_000_000_000	Your answers required
10	1 1_000_000_000_000	Your answers required

For the above question, you must provide/develop multiple solutions, provide complexity analysis of each solution, test cases used, results generated, and running time taken to generate the results. You must also provide the Python codes.

Deliverables

Deliverable 1 – the report in PDF format

A template will be provided. Your report must contain:

I.	Create a solution! Come up with ONE BASIC different way to solve the problem. Include a brief explanation of how you understood the problem, your approach to solving it, and a short description of your code. The complete codes, along with comments, need to be added to the report's Appendix, as well as uploaded for Deliverable 2.	[30%]
II.	Test and measure running your solution! Come up with test cases that demonstrate the correctness of your solution. Explain why you chose these particular test cases. You can present your results in a table, showing different inputs in the test cases along with their runtimes. You can create a graph to visualise how the runtime changes with different inputs.	[20%]
III.	Optimise your solution! Optimise the solution you created earlier in (I) and improve its efficiency. Share your reasons for all the optimising steps that you took. Include the final code with outputs and running time measurements in the report's Appendix section and also upload it for Deliverable 2. Remember, your optimisation goal is to solve	[20%]

	correctly as many test cases as possible, as shown in the test case table above.	
IV.	Compare the performance! Compare the performance of the optimised and the original solution you created. Focus on understanding their time complexities and big-O notations. Create a graph to visualise how their running time compares to the test cases. Make sure to include detailed outputs, running time measurements for as many test cases as possible, and a graphical representation of your results in your submission.	[15%]
V.	Reflecting on work! Your task is to write about the limitations you faced. Summarise accomplished items. Make sure your whole report is well-organised, uses clear language, and includes proper references. This will help us understand your work better!	[15%]

Deliverable 2 – Source code & test case input files (marks included within Deliverable 1. 2)

The Python source codes (one for each solution), inputs/test cases used for analysis, and others. They should be placed in a zip file which must be uploaded separately from the report.

You must implement your solution using the **Python** programming language. You may use any sample code provided in the course lectures and laboratories/tutorials as an aid, but make sure you reference any source code or tools used.

Unreferenced codes/contents may involve an investigation into an academic misconduct offence.

You are strongly advised to commence working on this coursework as it becomes available and contact the Module leader with your query as early as possible.

Additional guidelines and advice for the coursework will be provided separately.

Grading Criteria

Criteria for Assessment	80-100	70-79	60-69	50-59	40-49	30-39	0-29
Content, knowledge and understanding	Demonstrates exceptional systematic understanding of problem solving, computer programming and algorithmic performance. There is exceptional evidence of engagement with all key elements.	Demonstrates an excellent systematic understanding of problem solving, computer programming and algorithmic performance. There is also excellent evidence of engagement with all key elements.	There is a very good systematic understanding of problem solving, computer programming and algorithmic performance. There is also some very good evidence of engagement with all key elements.	Has demonstrated a good understanding problem solving, computer programming and algorithmic performance. There is also some good evidence of engagement with most key elements with some omission of detail.	Has demonstrated a satisfactory level of understanding of problem solving, computer programming and algorithmic performance. There are a few notable omissions and there is limited evidence of engagement with all key elements. Overall a satisfactory attempt at these criteria	A poor understanding of one or more of the following - problem solving, computer programming and algorithmic performance. There is insufficient evidence of engagement with the key elements. Overall an unsatisfactory attempt.	Little or no understanding of one or more of the following - problem solving, computer programming and algorithmic performance. There is very little evidence of engagement with the key elements. Overall a very unsatisfactory attempt.
Cognitive/Intellectual Skills	Demonstrates exceptional use of a critical analysis of information leading to the proposal of a robust and detailed solution. There is exceptional evidence of reflection that identifies the strengths and weakness of the approaches undertaken.	Demonstrates an excellent use of a critical analysis of information leading to the proposal of a robust and detailed solution. There is also excellent evidence of reflection and judgement based on the interpretation of the results obtained.	Demonstrates a very good use of a critical analysis of information leading to the proposal of a detailed solution. There is also some very good evidence of reflection and judgement based on the interpretation of the results obtained.	Demonstrates some good critical analysis of information leading to the proposal of a detailed solution. There are some exposed weaknesses of cognitive skills. There is also some good evidence of reflection and judgement based on the interpretation of the results obtained.	Has shown some satisfactory level of critical analysis of information. There is evidence of reflection and judgement based on the interpretation of the results obtained at a threshold pass level.	Has shown little use of techniques to undertake a critical analysis of information. The reflection and judgement based on the interpretation of results is weak and lacks detail.	Has shown little or no use of techniques to undertake a critical analysis of information. The reflection and judgement based on the interpretation of results is very weak and lacks detail.

Communication, Organisation and Presentation **Graduate Employability and Application of Skills**	Demonstrates exceptional use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and of an exceptional quality. There is exceptional evidence of the qualities of transferrable skills necessary for employment that required personal judgement and successful experimentation.	Demonstrates excellent use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and of an excellent quality. There is excellent evidence of the qualities of transferrable skills necessary for employment that required personal judgement and successful experimentation.	Demonstrates a very good use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and overall is very good. There is also very good evidence of the qualities of transferrable skills necessary for employment that required personal judgement and mostly successful experimentation.	There is good use of argument and language which communicates information to the target audience. The structure and flow of the report is mostly coherent and overall is good. There is also some good evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is mostly acceptable with some shortcomings in the grammar. The structure and flow of the report is barely acceptable with some presentation issues. There is also some evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is mostly at a substandard level. The structure and flow of the report is unacceptable with some presentation issues. There may also be little evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is at a substandard level. The structure and flow of the report is unacceptable with significant presentation issues. There may also be little/no evidence of the qualities of transferrable skills necessary for employment.
Referencing, sourcing, acknowledging and coverage	The exceptional use of appropriate references reflects clear and detailed understanding of the referenced works and its contents from a variety of sources.	The excellent use of appropriate references reflects clear and detailed understanding of the referenced works and its contents referenced works.	The use of references reflects a very good understanding of the cited work and its contents. Some references may not be the most recent.	The use of references reflects a good understanding of the cited work and its contents. Some references may not be the most recent or are taken from a narrow range of sources.	The use of references reflects a satisfactory understanding of the cited work and its contents. Some references may not be appropriate or the most recent or are taken from a narrow range of sources.	The use of references reflects a poor understanding of the cited work and its contents. The references may not be sufficient or appropriate or the most recent or are taken from a narrow range of sources.	Little or no cited work. The references may not be appropriate or the most recent.