Faculty of Engineering, Environment and Computing

4061CEM Programming and Algorithms 1

Assignment Brief

Module Title	Assignment Type	Cohort	Module Code
Programming and Algorithms 1	Individual	September	4061CEM

Coursework Title	Hand Out Date
Brute-Forcing a Password (Attempt 3)	24/05/2023

Lecturer	Due Date and Time
Dr Ian Cornelius	03/07/2023

Estimated Time	Coursework Type	Credit Value Assessed
20 Hours **Word Limit** Not Applicable	Individual Programming Assignment	10

Submission Arrangement: Aula

File Types Accepted: `docx pdf`

Marks and Feedback Date: 24/07/2023

Feedback Method: TurnItIn

Module Learning Outcomes Assessed:

- Understand basic theoretical aspects that apply to programming
- Design simple software to meet given specifications

Important Notice

If this is your *third (and final) attempt* at this piece of coursework, you are taking the feedback from your previous submission and improving upon it for this submission.

Preliminary Instructions

Tasks are to be undertaken in the Python programming language.

You will be expected to include comments in your code to explain the behaviour of your code and provide a justification of your algorithm selection. You are also strongly advised to test your code for compilation on a system other than your own, prior to submission. Non-compiling code will not pass, see the marking rubric for further information.

You will create a Coventry University GitHub **private** repository to store your source-code and manage version control of your work. Evidence of version control must be included in your regular commits to the repository over the period between the hand out date and due date.

Your eventual submission via TurnItIn will link to that repository which must include all of your source-code.

Task and Mark Distribution

The purpose of this assignment is to create an application whereby a user can insert a SHA256 hash via the command-line or from a file and then brute-force it to decrypt the password. The assignment has been split into a collection of mini-tasks to help you build up the overall application. A CSV file will be provided for this task with a selection of SHA256 hashes to test your brute-forcing algorithm. The plain-text password will be provided alongside the hash to assist you in the debugging and creation of your application.

The password that will be used to check your code will consist of numbers, letters and special characters that are typically found on a **UK-layout keyboard**. The password will also be no more than five characters long. You do not need to use a salt when encrypting the guessed password.

Assignment Tasks

To successfully complete this assignment you are required to complete the following tasks. Each task has a weight that is attributed towards a portion of the overall grade.

Remember, that each of these tasks will feed into a project (or piece of code) that will run cohesively.

Task 1: User Input (10%)

For this task you are required to accept a user input and store it in an appropriate data type. The hash(es) you accept will be used in the decryption function.

It will be expected that the user can insert either a single hash, or multiple hashes via the command line.

Task 2: File Input (20%)

For this task you will be asked to read a supplied CSV file that will consist of SHA256 hashes. The file may consist of a single hash, or multiple hashes, and it will be expected that these are stored in an appropriate data type.

Example CSV File

Task 3: Brute-Forcing the Hash(es) (60%)

For this task, you are expected to take the collection of hashes (either user-inserted, or from a file) and decrypt them.

Each step in the decryption method should be shown in the command line, with the final output of the *cracked* password and the total amount of time taken for the decryption process.

It will be expected to see that students' have used multiple commits with descriptive messages for each commit. It will also be expected to see that you have made comments in your code to explain the methodology of the function.

Task 4: Submission Guidelines (10%)

You will be expected to follow the submission guidelines, as outlined in the document below. Essentially, you are required to follow these rules:

1. Page One: Consists of a GitHub URL to the repository of your source-code
2. For each source-code file you have (`filename.py`), you need to provide:
    1. A single page, with the name of the file
    2. A single (or multiple) page(s) with the source-code in that file
3. The Python code file(s) must be submitted, the screenshots will not be accepted.

It is easier to visualise what is meant by these rules by looking at the example document provided below.

Submission Example

You will either be awarded zero marks, or full marks, depending on how you follow the guidelines.

Your source-code will be submitted to a plagiarism checker - so please ensure that any source-code acquired online is appropriately referenced. You are not to push or commit code to the GitHub repository after you have submitted your coursework. Timestamps will be checked, and if any changes made after the submission timestamp will not be marked.

Extensions and Deferrals

If you require an extension or deferral, you can find out more information at the following link:

Information on Extensions/Deferral

Remember: You must supply evidence with your application. Failure to do so may result in the extension/deferral not being approved.

Plagiarism and Academic Misconduct Policy

Ensure that you are familiar with the university guidance on Plagiarism and Academic Misconduct. Any work found to be in violation of the rules will be submitted for an academic misconduct case.

Marking Allocation

0 - 39	40 - 49	50 - 59	60 - 69	70+	80+
Work mainly incomplete and/or weaknesses in most areas.	Most elements completed; weaknesses outweigh the strengths.	Most elements are strong; minor weaknesses.	Strength in all elements.	Most work exceeds the standard expected.	All work substantially exceeds the standard expected.

Marking Rubric

Task	Fail	Third	Lower Second	Upper Second	First
1	No attempt made.	Some attempt has	A function exists, but	A function has been	A function has been

		been made, but it does not function as intended.	only takes a single hash input and does not store it.	implemented that stores a single hash input.	implemented that stores a single and multi-user input in an appropriate data type.
2	No attempt made.	Some attempt has been made, but it does not function as intended.	A function exists, but only takes a single hash from the file and does not store it.	A function has been implemented that stores a single hash from the file.	A function has been implemented that stores a single hash from the file.
3	No attempt made.	Some attempt has been made, but it does not function as intended.	A function exists, but is only able to decrypt a static hash.	The function is able to take a single hash from a user input or file and decrypt it.	The function is able to take multiple hashes from a user input or file and decrypt it.
4	Did not follow the guide-lines.	Did not follow the guide-lines.	Did not follow the guide-lines.	Did not follow the guide-lines.	Followed the guidelines correctly.

Notes

1. You are expected to use the Coventry University APA Referencing Style. For support and advice on this students can contact Centre for Academic Writing (CAW).
2. Please notify your registry course support team and module leader for disability support.
3. Any student requiring an extension or deferral should follow the university process as outlined here.
4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system.
5. If there are technical issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours and will be communicated via your Module Leader.
6. Collusion between students (where sections of your work are similar to the work submitted by other students in this or previous module cohorts) is taken extremely seriously and will be reported to the academic conduct panel. This applies to both coursework and exam answers.
7. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, any test conditions and that demonstrated in a coursework assignment may result in you having to undertake a Viva Voce in order to prove the coursework assignment is entirely your own work.
8. If you make use of the services of a proofreader in your work you must keep your original version and make it available as a demonstration of your written efforts.
9. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this university, except resits, where for the coursework, you maybe asked to rework and improve a previous attempt. This requirement will be specifically detailed in your assignment brief or specific course or module information. Where earlier work by you is citable, i.e. it has already been published/submitted, you must reference it clearly. Identical pieces of work submitted concurrently may also be considered to be self-plagiarism.