Project 1 (30 points)

Write a program that works as follows:

- Retrieves the file path from the user. An acceptable file extension is only a zip file.
- The zip file may be password protected, check this. If the zip file is password protected then locate the matching password in the password collection. You should also measure and present how long it took to find the password.
- The password will be in a .txt file. There will be about 10,000 passwords in the file.
- After decrypting the zip file, list all the files that are inside the zip file.
- Then generate a checksum for each file (sha 256).
- Then make a query to virustotal.com requesting for the evaluation of a given file hash. You should check the documentation on the virustotal's website on how to perform API queries, in what format you will receive a response.
- Based on the responses for each file, generate an aggregate simple report.
- Files with .txt and .pdf extensions that will potentially be inside the zip file should be searched to contain keywords: PESEL, password, and email addresses. The number of occurrences should be counted for each file and for each word separately. Each uniqe e-mail address should be listed too. This information should also be added to the report.
- Generate the sha-256 checksum for the report file. The generated hash should be saved in a separate .txt file named hash.txt
- To the zip file, pack all the files that were already there, the report and the checksum for the report. Secure the zip file with the password: *P4$$w0rd!*
- [this step is no longer performed in the code] The result zip file and source code should be uploaded to Teams before the due date of the handover.

Requirements:

- you should annotate the program with comments, so that it is easier to understand what each part of the code does.
- the project may use external libraries.
- Exceptions and all errors should be handled.
- For each execution’s step should be generated log record. All log records should be saved to file log.txt
- The project should be presented and defended in class.
- There may be refinements to the requirements in the following days based on student input, keep track of updates (max 1 week)

Extra points could be added (max 5 extra points) for extra feature (1 extra feature = 1 extra point).

Ideas for extra features:

- GUI (web application),
- Storing information in DB (also keeping information about previous hashes),
- The final file is protected by hybrid cryptography (symmetric and asymmetric keys)
- Checking metadeta of files and analyze them for some reason (you need to explain what you look for and why)
- Providing some steganography module (please prepare some multimedia file with hidden information to proof that the module is working)

Extra feature shound enhace user options (including user-friendliness) or security options. I’m very open for your ideas.

The deadline for submission of the solution is: **16.06 g. 7:59 p.m**. No work will be accepted after this date.

You can submit the project earlier and arrange to defend the project. This is welcome.