LAB7

1. Write a program that will write the minutes passed from the start, every x seconds, where x is random chosen at each iteraton (from the inteval [a, b] , where a, b are arguments). The program will run infinitely.

2. Write two functions to check if a number is prime, and check which of them is more time-efficient.

3. Write a function that will receive as parameters two strings representing file paths and will return True if the files are identical or False otherwise.

4. Write a script that receives a directory as argument and creates a JSON file with data about all the files in that directory. For each file, the following information will be displayed: file_name, md5_file, sha256_file, size_file (in bytes), time when the file was created (human-readable) and the absolute path to the file.

5. Write a function that receives two parameters: a_path and ext. The script will add all files from the a_path folder that have the extension ext to a zip archive named the.zip.

6. Write a script that writes the day of the week for the New Year Day, for the last x years (x is given as argument).

7. Write a script to simulate loto 6/49 draw (numbers extraction). The output should be a list of six numbers between 1 and 49 representing the winning combination.

8. Write a function that receives two parameters: a_path and to_hextract. If a_path is a valid zip archive and to_hextract is a file inside the arhive the function will return the md5 digest for unzipped content of to_hextract and None otherwise.

9. Ics company managed to successfully stop a DDOS attack.

But there is one last task: there is a suspicion that after the DDOS attack, some files were compromised. Those files were digitally signed before the attack, using a proprietary algorithm. We have provided the algorithm with which we can check the integrity of the files:

- The contents of the files are structured as follows: **<CONTENT>...</CONTENT><SIGNATURE><TYPE>md5|sha256|sha512</TYPE><HEXDIGEST>...</HEXDIGEST></SIGNATURE>**
- The content(bytes) between <CONTENT> and </CONTENT> tags is extracted. The key **b"\xd0\xf3\xde\x9a\x8c\x80\x8d\xf0\x92n7\x94"** is appended to this content and then the hashing method specified between <TYPE> and </TYPE> tags is applied. The hexdigest obtained must be identical with the one specified between <HEXDIGEST> and </HEXDIGEST> tags. If not, it eans that the file have been compromised.

Write a function that will receive a file path as parameter and will return True if the file was compromised, or False otherwise.

Samples to test your script : SAMPLES