CS251: Python Outlab: Thanos and the The Infinity Stones

- Please adhere to the submission guidelines. It carries a penalty of upto 5 marks for not following the submission guidelines.

Introduction

> *"Before creation itself, there were six singularities. Then the universe exploded into existence, and the remnants of these systems were forged into concentrated ingots... Infinity Stones."*
>
> – *The Collector*

Thanos is on the rise and wants to get all the Infinity Stones to achieve ultimate power. The Marvel heroes have realized this, but Thanos has already joined forces with famous supervillians and has acquired 4 of the 6 stones (thanks to Thor and Hulk, we have the *Reality Stone* and the *Soul Stone* in safe hands). The heroes now look up to you as the ultimate saviour of the multiverse and seek your help in finding the Infinity Stones. Let's go save the day!

1 Guardians of the Galaxy

"I am Groot."

– *Groot*

After a fierce battle with Yondu and the Ravagers, Star Lord has secured the vault containing *The Orb*, concealing the *Power Stone* which is present in the `GOTG` directory. But the catch is, it requires a key to be opened which has to be passed as an argument (for example, if the key is *12345*, then the command would be `./vault 12345`.
Yondu is extremely careful about his keys, and he also likes to access his resources remotely. The key can be obtained by logging into `this` page. The credentials to log in are present somewhere, lying "under the hood". Try to find out where the credentials are (you have already done a lab on HTML). Since Star-Lord is bad with code, Yondu has configured the page to accept ONLY those requests which have the correct credentials and which are sent using the `urllib` package of Python. Star-Lord now needs you help in getting the vault key.

- Write a python script `get_vault_key.py` which gets the **correct** response from the page and prints the contents into standard output.
- Also take the key and execute `./vault <key>`. Write the outputs of both your python script and the result of the vault in a file `result.txt` inside the `GOTG` directory.
- How do you think Yondu is able to "filter" requests based on whether they are sent from the `urllib` package or not? (Hint: HTTP headers). Write your inferences in `result.txt`.

1.1 Suggested reading

- HTTP requests
- `urllib` (Python 3)
- params and HTTP request headers

1.2 Submission

Submit your `get_vault_key.py` and `result.txt` files in the directory `GOTG`.

2 Doctor Strange

> *"Wise choice. You'll wear the Eye of Agamotto in time. Until then, best not to walk the streets with an Infinity Stone."*
>
> – *Wong to Doctor Strange*

Dormammu, the ultimate superpower, has joined forces with Thanos, and with the help of Mordo, stole the *Eye of Agamotto*. Mordo decides to scatter the parts of the *Eye of Agamotto* among different universes in the multiverse, which he has numbered starting from 1, and he wants to scatter it as widely as possible. But Mordo, being bad with numbers, comes up with the following sequence -

- The first number is 1.
- The next number is generated by concatenating the "current number's digits and their frequencies from left to right" (read more `here`).

For example, the first number in the sequence (1) has 'One' number of ones. The next number is therefore, 11 (One One). The second number (11) can be read as 'Two' ones. The next number is therefore, 21 (Two One). Here are some more examples of a current number and its next number in the sequence (with hyphens for breakdown of the numbers):

- 1312 : 11131112 (11-13-11-12)
- 311119 : 134119 (13-41-19)
- 1111111111 : 101 (10-1)

Dr. Strange has figured it out and has written the "spell" for opening the dimensions and retrieving the parts (in the file `retrieve.py`). But he needs your help to write the generator to generate the numbers.
You need to write a generator function which generates the sequence given above, in the file `generator.py` and run the file `retrieve.py` given in the directory `DoctorStrange`. The output is generated automatically from `retrieve.py` into `eye.txt`.

Note: DO NOT use for and while loops for this task (use only Pythonic ways). Using loops will result in penalties.

2.1 Suggested reading

- generators in Python (start `here`)
- itertools
- If you couldn't figure out the sequence, read `this`

2.2 Submission

Submit your `generator.py` and `eye.txt` files in the directory `DoctorStrange`.

3 Captain America

> *"My weapons contain enough destructive power to decimate every hostile capital on Earth. Quite simply, gentlemen, I have harnessed the power of the gods."*
>
> – *Johann Schmidt*

Jonann Schmidt had the Tessaract, but thanks to Captain America, you have the container and the Tessaract now. But the container is locked using Vibranium locks, and there is only one way to open the box, that is to get the key from Zola. You have the key, but unfortunately, you could only acquire a corrupt version of the key. Your intel says that he has simply processed the image to have very little contrast.

You need to write a Python program `decoder.py` which processes the numpy matrix present in `message.npy`. Use *linear contrast enhancement* (see Suggested Readings) to uncover the message. Use `matplotlib` to view the images. Save the result in `processed_image.png` (your code should do that automatically). Also write the key obtained in `key.txt`.

3.1 Suggested reading

- Numpy (Python)
- Contrast Stretching (follow the tutorial `here`, only the formula is enough).

3.2 Submission

Submit `decoder.py, processed_image.png, key.txt` inside the `CaptainAmerica` directory.

4 Avengers

"I don't know what this is. Not really. I know it's not of this world, that it powered Loki's staff, gave you your abilities. But its true nature is a mystery and yet it is part of me."

– Vision and Scarlet Witch

As soon as you were going to get hold of the Mind Stone, Loki pops out of nowhere and hides the key inside a text file given in a huge list of folders. You have to find out the place where the real key is hidden. Among all the `key.txt` files, only one contains the actual key. The correct file contains a message of the following format:

```
The key is:  <32_chars_long_string>
```

Write a Python script to search all the files inside the directory `World` and print the complete path to the correct `key.txt` file along with the contents of the `key.txt` file. Also, submit the correct `key.txt` file.

4.1 Suggested reading

- os, re module

4.2 Submission

Submit `searchbot.py, key.txt` inside the `Avengers` directory.

5 The final attack

5.1 (Bonus: only if you have solved all the other questions)

You have got all the 4 infinity stones (their hashes actually). Thanos is on the verge of defeat now, but there is the final message you must uncover before returning back to a normal, boring life. The final message is hidden in `final_message.zip` but it is password protected (obviously). The list of words among which the key lies is also scrambled.
Fortunately, you have a decrypter which will recover it if you give the keys of all the stones. Run the executable `decrypt`, follow the instructions (with `scrambled.txt` in the same folder), and you must get a decrypted version named `words.txt`. Now, write a Python script to perform a **dictionary attack** on the zip file provided to extract all the files in the zip file. Your script must also print the word that is the correct password, and print **nothing else**. Name the script `zipper.py`

Hint: You do not have to generate any new passwords, the password is one of the words in the decrypted text file.

5.2 Suggested reading

- Libraries like gzip, zipfile, etc.

5.3 Submission

Submit `zipper.py, final_message.txt` inside the `Bonus` directory.

6 Submission format

Please submit according to the following format:

```
lab7_rollno1_rollno2_rollno3/
├── readme.txt
├── GOTG/
│   ├── get_vault_key.py
│   └── result.txt
├── DoctorStrange/
│   ├── generator.py
│   ├── eye.txt
│   └── key.txt
├── CaptainAmerica/
│   ├── decoder.py
│   ├── processed_image.png
│   └── key.txt
├── Avengers/
│   ├── searchbot.py/
│   └── key.txt
└── Bonus
    ├── zipper.py
    └── final_message.txt
```