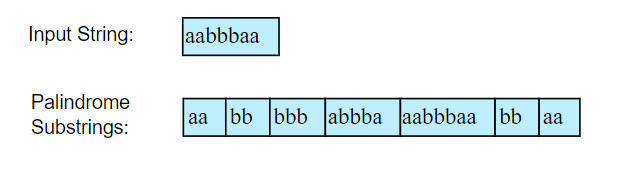
**Midterm2 Questions**

Day 09 Challenges:

**Challenge 1**: write a program that finds all nonsingle letter substrings that are palindromes. For instance:

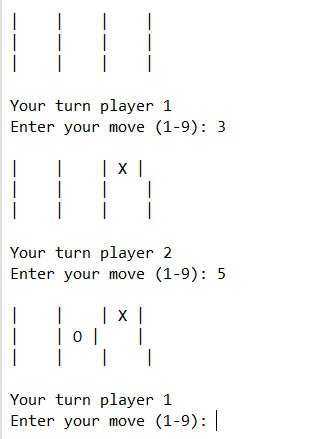


**Challenge 2**: there is a sorting algorithm called **Quick Sort** that used a recursive method. Learn how that works.

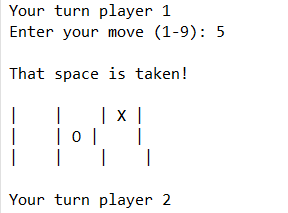
**Q:** write the power x to n function through recursive method. Assume n is positive.

**A:** Day09\_57.py

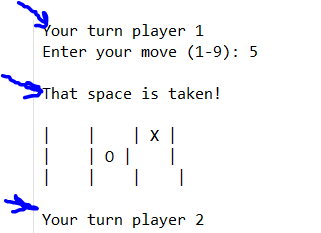
**Challenge 3**: create a tic-tac-toe game that its output is like this:



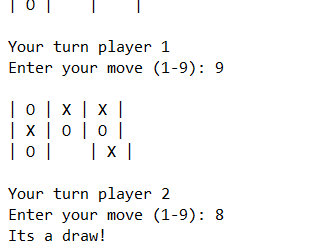
If the user enters a repeated cell it says:



If the user enters an already taken cell, it passes the turn to the next player and gives the following message:



And at the end it says the result:



Day 10 Challenges:

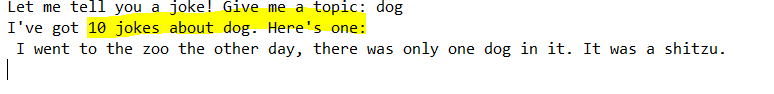
**Challenge 1:**

*The above module can be used to compare the big file sizes. For example, if you want to see if your 5 TB file is same as the one that your colleague has, you can check their message digest. Try to solve this problem. You can get some idea by looking at the code in this link but do not use* ***md5*** *and also do not use the file I/O methods that is suggested in this code (*[*https://stackoverflow.com/questions/36873485/compare-md5-hashes-of-two-files-in-python*](https://stackoverflow.com/questions/36873485/compare-md5-hashes-of-two-files-in-python)*)*

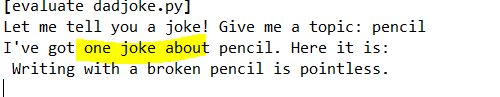
**Challenge 2:**

*Please read this link (https://leons.im/posts/a-python-implementation-of-simhash-algorithm/) and install simhash and test it with two files that are similar together and prove that you can find similar contents.*

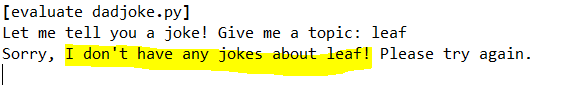
**Challenge 3**: use the information you have gained so far to create a program that asks the user for a topic and search the <https://icanhazdadjoke.com> and if it finds more than one jokes, then it randomly chooses one of those and shows it to the user.



If it finds one, it shows:



If it does not find any joke it says:



**Challenge 4**: Write a program to count a number of files for each extension in the given directory. The program should take a directory name as **argument** and print count and extension for each available file extension. For example, the output would be something like this:

python counter.py *\yourpathhere*

14 py

4 txt

1 csv

**Challenge 5**: familiarize yourself with the “**re**” module (<https://docs.python.org/3/library/re.html>). It is an abbreviation for the Regular Expression. They are used to find a string with a specific pattern. For example, the phone numbers in Toronto have 10 digits and they have this format xxx-yyy-wwww

Use the re module to check if the user has entered a valid phone number that follows the above-mentioned pattern.