The dataset When unzipped, the file is 2.8GB

Code the following Questions in python.

Question 1 (100pts) attached you will find a large file with a large number of repeating ASCII passwords (320,412,510 passwords) corresponding to 2,151,219 unique passwords. Based on the contents of the file and what you learned in the class, perform the following (using python):

1. **Frequency analysis:**
   1. Perform a frequency analysis of the passwords. The outcome of this analysis should be a sorted file with two columns. The first column in the resulting file is the number of times a password appears in the file, and the second column is the password itself. The second column is the set of unique passwords in the file.
   2. What do you conclude about the distribution of the passwords and their frequency? Comment on whether that is surprising or in line with your expectations.
2. **Strength analysis:** 
   1. Perform a strength analysis of the passwords in the file. The strength analysis should follow the Shannon entropy. Use the entire set of passwords in the file as your space of probability calculations. Here is an example of how the Shannon entropy works with the entire file as your calculation base. Let PWD = {Aziz, Mohaisen, is, a, har$h, professor} be your dictionary of passwords. First you calculate the probability for any character (case sensitive) to happen in any password (Aziz, Mohaisen, etc.). For example, all passwords have 29 characters in total, and A happens only once, so the probability for A is 1/29. Probability of a, on the other hand is 3/29, and so forth. Then, for a given password w, you construct a probability distribution over the characters, plug that distribution in the Shannon formula explained in the class, and done. The outcome should be a sorted file of two columns. The first column is entropy value, and the second column should be a unique password in the passwords dictionary (there is no point of including the same password multiple times, since they will have the same entropy).
   2. Comment on the strength analysis of the passwords and any obvious trends
3. **Strength analysis of unique passwords:**
   1. Redo question 2, but this time using the set of unique passwords in column 2 of question 1. This is, when calculating the probability as in question 2, consider your space of probability to be driven from the unique passwords dictionary. Compare the results with question 3.
   2. Comment on the comparison, and how the frequency of a password in a dictionary make that password a good (or a bad) choice with respect to security.
4. **Offline dictionary attacks:** The hash values (MD5 hashes, installed) below were revealed when a DB was compromised. Answer the following
   1. If you are to try to find the original passwords in those hashes, how would you approach the problem?
   2. If you are told that the passwords are already in the dictionary you are given, how your approach would be different?
   3. Find the passwords of those hashes, given that they are in the dictionary.
      1. ba931c15ec0163c4bb339f41571694ce
      2. c9cd905fc459e5550b8c3b01d4346c25
      3. e9269d9e52a692f52caece9d0e7cdae1
      4. 660719b4a7591769583a7c8d20c6dfa4
5. **Contexts for targeted password cracking:** Jim is a fan of the Buffalo Bills. This, in turn, is reflected in his choice of passwords. Often time, he would use his passwords as words related to the Buffalo Bills, including names of players, years of wins, etc. The MD5 (unsalted) hash value of Jim’s password is “83bfc234f88cc75d52e9ec24e54bc8be”. Answer the following:
   1. Provide 3 possible candidates for what the password could be
   2. If you are told that the password is a family name of a player playing for the Bills, how would this help you find the password?
   3. If you are told that the password is a family name of a player who retired on April 7th, how would this help you find the password?
   4. Use a side channel to infer the password, without knowing the context above (e.g., search Google).
   5. What does this tell you? About salting, using common passwords, etc.