Final Python Project: Easy

This is an alternative to the complete Final Project. It is an easier task working from the same data. The trick is that it can only lead to passing the course, not to a higher grade (for Python, which generally means for the course overall).

Portions of a gene have been sequenced for a group of patients. These have been compared to a standard reference via BLAST to identify specific mutations (SNPs). Two search hits have been found. We'll work with a simplified version of the analysis which still has enough complexity to act as a realistic problem but doesn't work well with automated tools.

Your data file holds the results for many patients, one after the other. A simplified example of the part of the sequence area for one patient appears below:

```
Query_1  241   ACCGTGCGTGGTTTTTCACAGTTCGGGGAT  270
HAP_A    281   .T........................T...  310
HAP_B    281   ...........C..............T...  310

Query_1  271   TCGTATGTGTATGTCATTTTTGAAGACGGC  300
HAP_A    311   ..............................  340
HAP_B    311   .............................G  340
```

The first line shown here is for the query (reference). It holds the query name, the starting position in the sequence, a portion of the sequence and the ending position for this fragment. The lines immediately following hold the aligned hits with similar fields except that matches to the query are replaced with "." (period) and the differences show the base code for the mutation.

A blank line separates portions of the sequence from each other until the sequence ends. There are other lines before and after the sequence data. These are not relevant for this study. Many patients are in this file but you will only analyze the first.

Looking at the data in this area, we see a total of two SNPs (mutations) for HAP_A and three for HAP_B. The positions of the differences can be read from the position in the line relative to the start (or finish) of the query (ignore the position numbers for the haps). For example, HAP_A shows C242T, where C is the base in the reference, T is the "mutation" and 242 is the position in the query (one away from the start, 241). Similarly, HAP_B shows C300G (not 340).

Task:

Print a list of the locations of all mutations found in the two search hits (haploids). These positions must be based on the Query sequence, not the haploids. The exact format of the output can be with or without commas doesn’t to be perfect, as long as I can read it and the answer is correct.

For example, your output may look like:

```
A: 267 339 345 393 396 405
B: 134 252 267 300 339 357
```

or

```
A: [267, 339, 345, 393, 396, 405]
B: [134, 252, 267, 300, 339, 357]
```

except that you will probably have different numbers and many more of them. I don't care about line wrapping but you can include it if you wish.

General Procedure:

The following is a list of suggested "subgoals" to help you. None are strictly required but some progression from simpler scripts to the final should be shown.

1. Write a script which extracts the first patient from the large file. Look through the file to determine how to identify the last line for a patient. Write this to a new file to be used as the input for the remainder of the project.

2. Identify lines which hold the query sequences. Get the current starting position. Get the mutations and keep the haploids separate.

3. Recall that blank lines separate groups of sequence lines (query and hits). Look for "blank" lines (allowing for a couple characters like newlines, etc).

Comment your code for readability. Show what you did and why – possibly in simple Notebook cells with comments. Create functions and use standard language features as appropriate. Cite anything taken from someone else (in class or external). Do not copy too much from other sources. This should primarily be your work.