Name: Rafael Wang
USC Student ID: 6189106477
Email: rafaelwa@usc.edu

Program Summary:
In this program (main.py), I use numerical descriptions based on biochemical properties alongside ACC as the features for a machine learning algorithm. Based on these features, I then use RFE for feature reduction and then use SVM techniques to predict a new multi-epitope vaccine sequence. I use the NegativeB.txt, NegativeT.txt, PositiveB.txt, and PositiveT.txt as my input files.

References:
I referenced https://github.com/zikunyang/DCVST, which contains the code used for the original study and some important .txt files that I use.

Assumptions:
Many key assumptions were made in this program. First, for each individual sequence, I set the max amino acid length as 10, since SVMs must work with constant fixed-length vectors.
Secondly, I randomly imported 180 positive B-cell epitopes and 180 negative B-cell epitopes to train the ML model. I randomly imported 150 positive CTL epitopes and 150 negative CTL epitopes to train the ML model. I also did 150 of each for the HTL epitopes.
I also assumed that sequences of length 11 or fewer amino acids were CTL epitopes while those greater than length 11 were HTL epitopes.
Of the selected sequences, I made 25% of them test sequences.
For predicting a new vaccine, I used the remaining sequences that were not used as part of training each of the models.
The new vaccine takes the top 10 epitopes from each of B-cell, HTL, and CTL epitopes (that were not used as part of training).

I/O:
The final constructed vaccine is both outputted to the terminal and outputted to a separate constructed_vaccine_sequence.txt file. The feature selections and results (including accuracies) for the B-cell, CTL, and HTL epitopes are all outputted to the terminal.

Results:
While the performance of the CTL epitope model was great and the HTL epitope model was acceptable, the B-cell epitope model was not as strong, despite me giving it a greater num_to_choose (of 180) as opposed to the 150 for the T-cells. I also tried making other modifications within the code, but the tradeoffs in terms of efficiency and runtime were quite large, so I recognize it is an area for future improvement and optimization.