Siqi Wen
HW6: Coin War

This program allows the user to input a starting position of Coin War, it takes input from a file named on the command line. Then the program will play the game to completion and report the game result. It will print “1” if Player 1 wins, “2” if Player 2 wins, or “0” if there is a tie.

Here are things I did to develop this program and how it would work:

1. Imported “random” and “sys” to access the random module and sys module;
2. Function “read_list” will read the file that is named on the command line and return a list of two strings. The sequence of the letters in each string will be either random or set based on what the first line says in the file. If the first line in the file says “random”, then the program will create two armies in random sequence for Player1 and Player2, and each army would contain five letters. In addition, Player1’s army and Player2’s army will be put into a list, which will be the return value for this function; if the first line in the file says “position”, then the sequence of these two armies will be the same as the ones in the second line and third line in the file.
3. Another function “coin_war” will take Player1’s army and Player2’s army as arguments and play the game by following the rules, then return 1 if player1 wins, 2 if player2 wins, or 0 if there is a tie. In the function, I created two empty lists for player1’s and player2’s prisoners. Then these armies lists and prisoners lists will make some changes in every step based on the game rules. I used “pop() method” to remove the first element in the lists; “append() method” to append the element at the end of the lists; “extend() method” to add the specified list elements to the end of the current list; “del operator” to delete the elements in the prisoners lists after the program moved the elements in the prisoners lists to player1’s or player2’s army list. The important rule is that the game is immediately over when either player has no army at any point — When the first elements in player1’s and player2’s armies are the same, after each player puts their first coin at the end of their prisoners, the program needs to make sure that player1 and player2 both still have elements in their armies before the same procedure happens again.
4. In general, the program will print the starting position of player1’s army and player2’s army, then print “1”, “2” or “0” depending on the game result.

In the beginning, I was very confused about this homework. But after I asked our TA Mike and Professor Bart for help and re-watched Professor Bart’s videos, I finally figured out how to develop this program, I spent about 4 hours to write the code, and I felt very happy when I tried my program with the test files and the results were all correct. There was one bug that I encountered while trying out my program was that I forgot to delete the elements in the prisoners lists after all the elements in them have been moved to player1’s or player2’s army list, I’m glad I was able to fix the bug after I read the rule again and spent more time thinking through the problem. Overall, I’m very happy that I was able to develop this program and I really think it’s a great way to practice what we’ve learned!