Assignment 4 - James Byrne - 16310943

To test the readFile function, the following tests were done

readNonExistingFile_testcase
The assertion works and the function prints an error when run as the file or directory doesn't exist. I've tested this as it is possible that a file path is put into the function that doesn't exist.

readEmptyFile_testcase
This failed which is expected as there are no players in the file. I checked this as it is possible that the file doesn't contain anything.

readTenFile_testcase
This works, which is as expected as this is a valid players file with ten players. I tested this as this a valid players file to ensure the function worked correctly.

readElevenPlayers_testcase
This works no problem although when loop in the function reaches the end of the array it continues to read data from the file and stores it in space not allocated to the array. I tested this as it is possible that there are more players in the file than there are elements in the array.

To test the sortPlayers function, the following tests were done

sortZeroPlayers_testcase
This works as no elements in the array is changed, so the expected result is that the array of players is the same before the array is sorted. This was tested as it is a border case to see whether the array is unchanged.

sortOnePlayer_testcase
This works as the size of the array to be sorted is one, so no arrays can swap as there is only one. This is tested as it is a border case to see whether the array is unchanged.

sortTenPlayers_testcase
This doesn't work, which is not what is expected as entering size 10 with an array of players of size 10 is valid. This was tested to check whether the function does what it is meant to do.

sortLargerSize_testcase
This doesn't work which is expected as the garbage values in the memory after the array are swapped with ones in the array leaving garbage values in the array and players outside of the array. This was tested in case there was a mismatch with the number of players and the number of elements in the player array where the size is larger than the elements in the array.

sortSmallerSize_testcase
This doesn't work which is expected as the elements in the array that are greater than the size imputed e.g. size 6, element 7 is greater than size. This was tested in case there was a mismatch with the number of players and the number of elements in the player array where the size is smaller than the elements in the array.

To test the shufflePlayers function, the following tests were done

shuffleZeroPlayers_testcase
This works as is expected as the size of the array put in as a parameter is zero so no shuffling will happen. This results in the playerArray being unchanged. Hence the Assert is false as the strings are equal because the array wasn't shuffled. This was tested as it is a border case.

shuffleOnePlayers_testcase
This works as is expected as the size of the array put in as a parameter is one so no shuffling will happen as at least 2 elements are required for shuffling to work. This results in the playerArray being unchanged. Hence the Assert is false as the strings are equal because the array wasn't shuffled. This was tested as it is a border case.

shuffleTenPlayers_testcase
This doesn't work, which is not expected as this case is a valid case to test whether the function is doing what it is meant to be doing. This test was done to check whether the function operates correctly

shuffleElevenPlayers_testcase
This also doesn’t work which is expected, but due to different reasons as the function is flawed. This was tested in case there was a mismatch with the number of players and the number of elements in the player array where the size is larger than the elements in the array.

shuffleLargerSize_testcase
This doesn’t work due to the fact that the function doesn't do what it is meant to do. This was tested in case there was a mismatch with the number of players and the number of elements in the player array where the size is smaller than the elements in the array.