Functions	#	Test Description	Test input	Expected output	Actual output	P/F
initializedPlayers	**1**	Initializes the values of the struct player array to default values	N/A	Initializes the values of the struct player array to default values	Initializes the values of the struct player array to default values	**P**
initializedPets	**1**	Initializes the values of the struct pet array to default values	N/A	Initializes the values of the struct pet array to default values	Initializes the values of the struct pet array to default values	**P**
getComPetDium - Uploads the BattlePets from an existing file (ex. ComPetDium.txt) to the BattlePet array	**1**	file contains correct layout of pet details	Solaraffe Fire A fiery and majestic giraffe 0 Firepupp Fire A loyal, flame-breathing dog 10	dCurrentPets = 2 pet[0].name: Solaraffe pet[0].affinity: Fire pet[0].description: A fiery and majestic giraffe pet[0].matchCount: 0 pet[1].name: Firepupp pet[1].affinity: Fire pet[1].description: A loyal, flame-breathing dog pet[1].matchCount: 10	dCurrentPets = 2 pet[0].name: Solaraffe pet[0].affinity: Fire pet[0].description: A fiery and majestic giraffe pet[0].matchCount: 0 pet[1].name: Firepupp pet[1].affinity: Fire pet[1].description: A loyal, flame-breathing dog pet[1].matchCount: 10	**P**
	2	file contains incorrect layout of pet details	Solaraffe 123 741908749 abc !!!%%&&**$() \\1]2]3] A loyal, flame-breathing dog ##	dCurrentPets = 0 "Invalid pet file. Please use the correct pet template."	dCurrentPets = 4 pet[0].name: Solaraffe pet[0].affinity: 123 pet[0].description: 741908749 pet[0].matchCount: 0 pet[1].name: bc pet[1].affinity: !!!%%&&**$() pet[1].description: \\1]2]3] pet[1].matchCount: 0 pet[2].name: loyal, pet[2].affinity: flame-breathing pet[2].description: dog pet[2].matchCount: 0 pet[3].name: # pet[3].affinity: pet[3].description: pet[3].matchCount: 0	**F**
	3	file is not found	[empty file]	dCurrentPets = 0 "File not found."	dCurrentPets = 0 "File not found."	**P**
	4	file is empty	[no file]	dCurrentPets = 0 "No pets found in the file."	dCurrentPets = 0 "No pets found in the file."	**P**
getPlayers - Uploads the players from the file Players.txt	**1**	file contains correct layout of player details	_Player1_ crimenology 0 1 2 Player2%^&* emerald 4 4 0	dCurrentPlayers = 2 players[0].name: _Player1_ players[0].savedPassword: crimenology players[0].wins: 0 players[0].loss: 1 players[0].draws: 2 players[1].name: Player2%^&* players[1].savedPassword: emerald players[1].wins: 4 players[1].loss: 4 players[1].draws: 0	dCurrentPlayers = 2 players[0].name: _Player1_ players[0].savedPassword: crimenology players[0].wins: 0 players[0].loss: 1 players[0].draws: 2 players[1].name: Player2%^&* players[1].savedPassword: emerald players[1].wins: 4 players[1].loss: 4 players[1].draws: 0"	**P**
	2	file is not found	[empty file]	dCurrentPlayers = 0 "File not found."	dCurrentPlayers = 0 "File not found."	**P**
	3	file is empty	[no file]	dCurrentPlayers = 0 "No players found in the file."	dCurrentPlayers = 0 "No players found in the file."	**P**
getTxtname - Gets the player name and returns it as a .txt file format	**1**	normal single word Player name	Input: Player1 Prefix: saved_roster/	Expected: saved_roster/Player1.txt	Output: saved_roster/Player1.txt	**P**

	2	Player name with spaces	Input: John Doe Prefix: saved_roster/	Expected: saved_roster/John Doe.txt	Output: saved_roster/John Doe.txt	**P**
	3	empty Player name	Input: Prefix: saved_roster/	Expected: saved_roster/.txt	Output: saved_roster/.txt	**P**
	4	different prefix	Input: Player2 Prefix: match_	Expected: match_Player2.txt	Output: match_Player2.txt	**P**
updatePlayerTxt - Updates the Wins Lost Draws of all players in players.txt	**1**	multiple Player data	players[0].name: Player1 players[0].savedPassword: password123 players[0].wins: 5 players[0].loss: 2 players[0].draws: 1 players[1].name: Player2 players[1].savedPassword: securepass players[1].wins: 3 players[1].loss: 4 players[1].draws: 2 players[2].name: Player3 players[2].savedPassword: mypassword players[2].wins: 7 players[2].loss: 1 players[2].draws: 0	Expected Player[0]: Name: Player1 Password: password123 Wins: 5 Losses: 2 Draws: 1 Expected Player[1]: Name: Player2 Password: securepass Wins: 3 Losses: 4 Draws: 2 Expected Player[2]: Name: Player3 Password: mypassword Wins: 7 Losses: 1 Draws: 0	Actual Player[0]: Name: Player1 Password: password123 Wins: 5 Losses: 2 Draws: 1 Actual Player[1]: Name: Player2 Password: securepass Wins: 3 Losses: 4 Draws: 2 Actual Player[2]: Name: Player3 Password: mypassword Wins: 7 Losses: 1 Draws: 0	**P**
	2	single Player data	players[0].name: SoloPlayer players[0].savedPassword: solo123 players[0].wins: 10 players[0].loss: 0 players[0].draws: 0	Expected Player[0]: Name: SoloPlayer Password: solo123 Wins: 10 Losses: 0 Draws: 0	Actual Player[0]: Name: SoloPlayer Password: solo123 Wins: 10 Losses: 0 Draws: 0	**P**
	3	Player name with spaces	players[0].name: John Doe players[0].savedPassword: password123 players[0].wins: 5 players[0].loss: 2 players[0].draws: 1	Expected Player[0]: Name: John Doe Password: password123 Wins: 5 Losses: 2 Draws: 1	Actual Player[0]: Name: John Password: Doe Wins: 1634755954 Losses: 1969448307 Draws: 0	**F**
updatePetscount - Searches for the pets used by the players and updates their match count	**1**	Player uses two pets	Player uses Flamepaw and Zapcat Flamepaw=0, Zapcat=0, Chillizard=0, Rocky=0, Aquafox=0	Player uses Flamepaw and Zapcat Expected: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=0	Player uses Flamepaw and Zapcat Actual: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=0	**P**
	2	Player uses same pet twice	Player uses Aquafox twice Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=0	Player uses Aquafox twice Expected: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=2	"Player uses Aquafox twice Actual: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=2"	**P**
	3	Player uses no pets	Player uses no pets Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=2	Player uses no pets Expected: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=2	Player uses no pets Actual: Flamepaw=1, Zapcat=1, Chillizard=0, Rocky=0, Aquafox=2	**P**

updateCompetdiumTxt - Overwrites saved info in competdium.txt to update match count		normal Pet data	pet[0].name: Flamepaw pet[0].affinity: Fire pet[0].description: A fiery creature that burns everything in its path. pet[0].matchCount: 5 pet[1].name: Zapcat pet[1].affinity: Electric pet[1].description: A cat that generates electricity to shock its enemies. pet[1].matchCount: 3 pet[2].name: Chillizard pet[2].affinity: Ice pet[2].description: A lizard that freezes its enemies with icy breath. pet[2].matchCount: 7	Expected Pet[0]: Name: Flamepaw Affinity: Fire Description: A fiery creature that burns everything in its path. Match Count: 5 Expected Pet[1]: Name: Zapcat Affinity: Electric Description: A cat that generates electricity to shock its enemies. Match Count: 3 Expected Pet[2]: Name: Chillizard Affinity: Ice Description: A lizard that freezes its enemies with icy breath. Match Count: 7	Actual Pet[0]: Name: Flamepaw Affinity: Fire Description: A fiery creature that burns everything in its path. Match Count: 5 Actual Pet[1]: Name: Zapcat Affinity: Electric Description: A cat that generates electricity to shock its enemies. Match Count: 3 Actual Pet[2]: Name: Chillizard Affinity: Ice Description: A lizard that freezes its enemies with icy breath. Match Count: 7	**P**
		empty Pet data	pet[0].name: pet[0].affinity: pet[0].description: pet[0].matchCount:	[unchanged]	[unchanged]	**P**
		Pet data with special characters	pet[0].name: Pet#1 pet[0].affinity: Water pet[0].description: A pet with special abilities! Can swim & dive. pet[0].matchCount: 10 pet[1].name: Pet_2 pet[1].affinity: Earth pet[1].description: A strong pet that can lift heavy objects. pet[1].matchCount: 15	Expected Pet[0]: Name: Pet#1 Affinity: Water Description: A pet with special abilities! Can swim & dive. Match Count: 10 Expected Pet[1]: Name: Pet_2 Affinity: Earth Description: A strong pet that can lift heavy objects. Match Count: 15	Actual Pet[0]: Name: Pet#1 Affinity: Water Description: A pet with special abilities! Can swim & dive. Match Count: 10 Actual Pet[1]: Name: Pet_2 Affinity: Earth Description: A strong pet that can lift heavy objects. Match Count: 15	**P**
startBattle	**1**	This function is responsible for the main gameplay of the program	N/A	This function is responsible for the main gameplay of the program	This function is responsible for the main gameplay of the program	**P**
displayRoster	**1**	Displays 3x3 player roster	N/A	Displays 3x3 player roster	Displays 3x3 player roster	**P**
displayChoices	**1**	Displays the choices for player selection the user can choose from	N/A	Displays the choices for player selection the user can choose from	Displays the choices for player selection the user can choose from	**P**
newPlayer - Creates a new player and saves it to the player array	**1**	normal Player name and password	lobster lobster1234	Expected: lobster 1234lobster 0 0 2	Actual: lobster 1234lobster 0 0 2	**P**
	2	Player name with special characters	P3P$R_G@ dlsu	Expected: P3P$R_G@ dlsu 0 0 0	Actual: P3P$R_G@ dlsu 0 0 0	**P**
	3	Player name and password with spaces	John Doe password1234	Expected: John Doe password1234 0 0 0 0	Actual: John Doe 0 0 0 0	**F**

loadSavedRoster - Loads the saved player roster txt file of a player	**1**	with existing player roster file	Ash_KetchEmAll.txt Zaprat Hydrake Terrashark Joltaroo Crocodice Steelephant Firepupp Solaraffe Zephyram	Ash_KetchEmAll Expected: pet[0].name = Zaprat pet[1].name = Hydrake pet[2].name = Terrashark pet[3].name = Joltaroo pet[4].name = Crocodice pet[5].name = Steelephant pet[6].name = Firepupp pet[7].name = Solaraffe pet[8].name = Zephyram	Ash_KetchEmAll Actual: pet[0].name = Zaprat pet[1].name = Hydrake pet[2].name = Terrashark pet[3].name = Joltaroo pet[4].name = Crocodice pet[5].name = Steelephant pet[6].name = Firepupp pet[7].name = Solaraffe pet[8].name = Zephyram	**P**
	2	missing player roster file	[no file]	"File not found"	"File not found"	**P**
selectPlayer - Selects a player from struct player array and sets it as the current player if their password is correct	**1**	select pets for player's roster	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	**P**
	2	load empty roster	[empty lobster.txt]	"No saved roster"	"No saved roster"	**P**
	3	give incorrect password	lobster lobster456 (should be lobster123)	"Incorrect password"	"Incorrect password"	**P**
selectPets - Lets existing users select their pets for the match	**1**	valid pet selection	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	1 Zaprat 2 Hydrake 3 Terrashark 4 Joltaroo 5 Crocodice 6 Steelephant 7 Firepupp 8 Solaraffe 9 Zephyram	**P**
	2	pet already chosen	[Zaprat already chosen] Zaprat	"Pet already chosen"	"Pet already chosen"	**P**

determineWinner	**1**	Determines and returns the winner of the match	Player1 = "Grass";
			Player2 = "Fire";
				2 wins	2 wins	**P**

computeBattle	**1**	Matches up the pets of the two players and stores the winners of each battle to the matchResult array	Player 1: {"Flamepaw", "Fire"}, {"Zapcat", "Electric"}, {"Chillizard", "Ice"} Player 2: {"Aquafox", "Water"}, {"Rocky", "Earth"}, {"Grasshopper", "Grass"}	Match results: ['2', '1', '1'] Flamepaw (Fire) vs Aquafox (Water) → Player 2 wins ('2') Zapcat (Electric) vs Rocky (Earth) → Player 1 wins ('1') Chillizard (Ice) vs Grasshopper (Grass) → Player 1 wins ('1')	Match results: ['2', '1', '1'] Flamepaw (Fire) vs Aquafox (Water) → Player 2 wins ('2') Zapcat (Electric) vs Rocky (Earth) → Player 1 wins ('1') Chillizard (Ice) vs Grasshopper (Grass) → Player 1 wins ('1')	**P**
displayMatch	**1**	Displays the match between the pets of the two players and the winner of the match	N/A	Displays the match between the pets of the two players and the winner of the match	Displays the match between the pets of the two players and the winner of the match	**P**
displayResult	**1**	Displays the match results in a 3x3 roster grid	N/A	Displays the match results in a 3x3 roster grid	Displays the match results in a 3x3 roster grid	**P**
checkLuckywin - Checks if the matchResult 3x3 has horizontal, diagonal, or vertical pattern	**1**	horizontal lucky win	matchResult.result = {'1', '1', '1', '2', 'D', '2', 'D', '2', 'D'};	isLucky = 1 (Lucky win detected) dWinner = 1 (Player 1 is the lucky winner)	isLucky = 1 (Lucky win detected) dWinner = 1 (Player 1 is the lucky winner)	**P**
	2	vertical lucky win	matchResult.result = {'1', '2', 'D', '1', '2', 'D', 'D', '2', 'D'};	isLucky = 1 (Lucky win detected) dWinner = 2 (Player 2 is the lucky winner)	isLucky = 1 (Lucky win detected) dWinner = 2 (Player 2 is the lucky winner)	**P**
	3	diagonal lucky win	matchResult.result = {'1', '2', 'D', 'D', '1', '2', 'D', 'D', '1'};	isLucky = 1 (Lucky win detected) dWinner = 1 (Player 1 is the lucky winner)	isLucky = 1 (Lucky win detected) dWinner = 1 (Player 1 is the lucky winner)	**P**
	4	no lucky win	matchResult.result = {'1', '2', 'D', 'D', '1', '2', 'D', '2', '1'};	isLucky = 0 (No lucky win) dWinner = -1 (No winner)	isLucky = 0 (No lucky win) dWinner = -1 (No winner)	**P**

typeOfwin
- Identifies what type of win it is
		lucky win	matchResult.result = {'1', '1', '1', 'D', 'D', 'D', 'D', 'D', 'D'};
			count1 = 3;
			count2 = 0;
				winType = "Lucky Win"	winType = "Lucky Win"	**P**

		majority win	matchResult.result = {'1', '1', '2', '1', 'D', 'D', '2', 'D', 'D'};
			count1 = 3;
			count2 = 2;
				winType = "Majority Win"	winType = "Majority Win"	**P**

		draw	matchResult.result = {'D', 'D', 'D', 'D', 'D', 'D', 'D', 'D', 'D'};
			count1 = 0;
			count2 = 0;
				winType = "Draw"	winType = "Draw"	**P**

returnWinner	**1**	Returns the winner of the match	N/A	Returns the winner of the match	Returns the winner of the match	**P**
createMatchHistory	**1**	Creates a new txt file which contains the previous match result	N/A	Creates a new txt file which contains the previous match result	Creates a new txt file which contains the previous match result	**P**
startComPetDium	**1**	This function is responsible for the manipulation of the battlepets	N/A	This function is responsible for the manipulation of the battlepets	This function is responsible for the manipulation of the battlepets	**P**
viewBattlepets	**1**	Displays all existing pets in pet struct	N/A	Display all existing pets in game	Display all existing pets in game	**P**
addBattlepet - Asks the user if they want to add one pet, add multiple pets from the import_pets folder, or exit		add one or multiple	1 add one	redirect to addOne function	redirect to addOne function	**P**
	2	exit	0 exit	exits	exits	**P**
	3	past choice range	9	"Invalid response. Please try again."	"Invalid response. Please try again."	**P**
editBattlepet - Asks the user which pet and what characteristic they want to modify	**1**	choose a pet to edit	Flamepaw	redirect to editBPname/editBPdetails/editBPdescription function	redirect to editBPname/editBPdetails/editBPdescription function	**P**
	2	exit	0 exit	exits	exits	**P**
	3	past choice range	9	"Invalid response. Please try again."	"Invalid response. Please try again."	**P**
deleteBattlepet - Asks the user which pet they want to delete from the ComPetDium	**1**	choose a pet to delete		redirect to deleteBattlepetDetails function		
	2	exit				
	3	past choice range				
saveRoster	**1**	Creates and saves or edits a 3x3 roster for a player	saved_roster/lobster.txt {"Flamepaw", "Fire"}, {"Zapcat", "Electric"}, {"Chillizard", "Ice"}	saved_roster/lobster.txt Flamepaw Zapcat Chillizard	saved_roster/lobster.txt Flamepaw Zapcat Chillizard	
isTxtFile	**1**	Checks if the given file is a .txt file.				
listTxtFiles	**1**	Lists all the existing .txt files in a folder				
addPetsfromExpansion	**1**	Adds one or multiple pets from petExpansion.txt to the ComPetDium				
editBPname	**1**	Edits a BattlePet's name and asks the user what they want to replace it with.				
editBPaffinity	**1**	Edits a BattlePet's affinity and asks the user what they want to replace it with				
editBPdesc	**1**	Edits a BattlePet's description and asks the user what they want to replace it with				
deleteBattlepetDetails	**1**	Deletes the details of an existing pet in competdium.txt and its structure in the BattlePets array				
checkIfPetMax	**11**	Checks if the number of BattlePets to be added will make the ComPetdium reach its max amount.				
addOnePetDetails	**1**	Adds one pet manually to competdium.txt and to the BattlePets array				

addMultiplePetsDetails	**1**	Adds multiple pets from a struct BattlePet array to the main BattlePets array and to the competdium.txt				
addOnePet	**1**	Aids the user in creating one BattlePet and adding it manually to competdium.txt and to the BattlePets array				
addMultiplePets	**1**	This function aids the user in adding multiple pets at once by selecting a file in the import_pets folder				
saveRosterToFile	**1**	Saves the current player's roster to a their respective file in the saved_roster folder				
viewStatistics	**1**	This function is responsible for the manipulation of the battlepets				
rearrangePets	**1**	Rearranges the pets from highest matchCount to lowest				
displayTop5pets	**1**	Displays the top pets used up to the rank the user wants to view				
rearrangePlayers	**1**	Rearranges the players from highest wins to lowest				
displayTop5players	**1**	Displays the top players				