Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
getDate	1	Getting the Date for May 30, 2022	n/a	returns nDay = 30, nMonth = 5, nYear = 2022;	returns nDay = 30, nMonth = 5, nYear = 2022;	P
	2	Getting the Date for June 1, 2022	n/a	returns nDay = 1, nMonth = 6, nYear = 2022;	returns nDay = 1, nMonth = 6, nYear = 2022;	P
inputTrip	1	User enters a valid Trip Number	user inputs "AE150"	tripNum = "AE150", *index = 9	tripNum = "AE150", *index = 9	P
	2	User enters an invalid Trip Number	user inputs "eeeee"	terminal prints "[INPUT ERROR] Invalid Trip Number, please try again.", function runs again	terminal prints "[INPUT ERROR] Invalid Trip Number, please try again.", function runs again	P
	3	User enters a string that is larger than 5 characters	user inputs "AE1501"	terminal prints "[INPUT ERROR] Invalid Trip Number, please try again.", function runs again	terminal prints "[INPUT ERROR] Invalid Trip Number, please try again.", function runs again	P
inputIDNum	1	User enters an 8-digit string input	user inputs "02115681"	returns IDNum = 02115681	returns IDNum = 02115681	P
	2	User enters a less then 8-digit string input	user inputs "1234567"	prints "[INPUT ERROR] Invalid ID Number Size, please try again" and function runs again	prints "[INPUT ERROR] Invalid ID Number Size, please try again" and function runs again	P
	3	User enters more than 8-digit string input	user inputs "123456789"	prints "[INPUT ERROR] Invalid ID Number Size, please try again" and function runs again	prints "[INPUT ERROR] Invalid ID Number Size, please try again" and function runs again	P
	4	User enters an input with 8 characters but it contains an invalid input	user inputs "1234e678"	prints "[INPUT ERROR] ID Number contains invalid data type, please try again" and function runs again	prints "[INPUT ERROR] ID Number contains invalid data type, please try again" and function runs again	P
inputName	1	User enters correct input	user inputs "Luis Miguel Antonio" and "Razon"	fullName.fullname = "Luis Miguel Antonio Razon"	fullName.fullname = "Luis Miguel Antonio Razon"	P
	2	User enters non-letter character in the first name input	user inputs "@Luis Miguel Razon"	prints "[INPUT ERROR] Name can only contain letters (A-Z, a-z)" and function runs again	prints "[INPUT ERROR] Name can only contain letters (A-Z, a-z)" and function runs again	P
	3	User enters more than 50 characters in an input	user inputs "Luis Miguel Antonio Luis Miguel Antonio Luis Miguel Antonio" and "Razon"	prints "[INPUT ERROR] First Name / Last Name cannot exceed 50 characters" and function runs again	prints "[INPUT ERROR] First Name / Last Name cannot exceed 50 characters" and function runs again	P
	4	User enters input with incorrect capitalization	user inputs "lUis miGuel anToNio" and "raZon"	fullName.fullname = "Luis Miguel Antonio Razon"	fullName.fullname = "Luis Miguel Antonio Razon"	P
inputPrio	1	User enters an input within the range 1 - 6	user inputs "1"	nPrio = 1	nPrio = 1	P
	2	User enters an input out of range	user inputs "7"	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-6." and asks for another input	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-6." and asks for another input	P
	3	User inputs an invalid data type	user inputs "abc"	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-6." and asks for another input	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-6." and asks for another input	P
getEmbarkationPoint	1	The function gets passed a value of less than 9	user inputs trip number: "AE109" resulting in tripIndex = 8	function returns 1	function returns 1	P

	2	The function gets passed a value of more than 9 but less than 19	user inputs trip number: "AE160" resulting in tripIndex = 19	function returns 2	function returns 2	P
inputDropPoint	1	embarkPoint = 1 and user inputs an integer between 1 and 3	user inputs "2"	function returns 2	function returns 2	P
	2	embarkPoint = 1 and user enters an integer less than 1 or more than 3	user inputs "4"	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-3." and asks for another input	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-3." and asks for another input	P
	3	embarkPoint = 2 and user enters an integer between 1 and 4	user inputs "4"	function returns 4	function returns 4	P
	4	embarkPoint = 2 and user enters an integer less than 1 or more than 4	user inputs "5"	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-4." and asks for another input	prints "[INPUT ERROR] Invalid Input. Allowed inputs are integers 1-4." and asks for another input	P
assignTrip	1	trip is empty	EmbarkCard user and trip[0].filled = 0	user is assigned to trip[0]	user is assigned to trip[0]	P
	2	trip has elements in it	EmbarkCard user and trip[index].filled = 1 (index = 0 to 3)	user is assigned to the trip[4]	user is assigned to the trip[4]	P
countEmpty	1	Only one of the elements is filled	trip[0].filled = 1	function returns 15	function returns 15	P
	2	Some of the elements is filled	trip[i].filled = 1 (i = 0 to 5)	function returns 10	function returns 10	P
	3	All of the elements is filled	trip[i].filled = 1 (i = 0 to 15)	function returns 0	function returns 0	P
getLowestPrio	1	The trip array is empty	trip[i].nPrio = 7 (i = 0 to 15, 7 is default value)	function returns 7 and i = 0	function returns 7 and i = 0	P
	2	The trip array's lowest priority is 4	trip[i].nPrio for i = 0 to 7, nPrio: 1 2 3 4 4 3 2 1	function returns 4 and i = 3	function returns 4 and i = 3	P
	3	The array is not full and the filled priority is 4	trip[i].nPrio for i = 0 to 15, nPrio: 1 2 3 4 4 3 2 1 7 7 7 7 7 7 7 7	function returns 7 and i = 8	function returns 7 and i = 8	P
moveUser	1	One of the elements of the array is empty	destination[i] is empty	function returns an empty array and destination[i] = user	function returns an empty array and destination[i] = user	P
	2	The array is full but one of the elements has a lower priority value	destination array is full but destination[i] has a lower priority value	temp = destination[i] and destination[i] = user and return temp	temp = destination[i] and destination[i] = user and return temp	P

sort	1 Integers are in increasing order already	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	P
	2 Integers are in decreasing order	trip[n].nPrio for n = 0 to 5 contains {6,5,4,3,2,1}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	P
	3 Integers are in random order	trip[n].nPrio for n = 0 to 5 contains {2,1,5,3,4,6}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	trip[n].nPrio for n = 0 to 5 contains {1,2,3,4,5,6}	P
copyTrip	1 Trip contains 1 element	source[0] is filled	dest[0] is filled, source[0] = dest[0]	dest[0] is filled, source[0] = dest[0]	P
	2 Trip contaiins no elements	source[i] is not filled, where i is 0 - 15	dest[i] is not filled, source[i] = dest[i] where i is 0 - 15	dest[i] is not filled, source[i] = dest[i] where i is 0 - 15	P
	3 Trip contains more than 1 element	source[i] is filled, where i is 0 - 15	dest[i] is filled, source[i] = dest[i] where i is 0 - 15	dest[i] is filled, source[i] = dest[i] where i is 0 - 15	P
printTripInfo	1 Trip has no passengers	passenger[i].filled = 0, where i is 0 - 15	prints "[O][O][O]\n[O][O][O]\n[O][O][O]\n[O][O][O]\n[O][D][D]"	prints "[O][O][O]\n[O][O][O]\n[O][O][O]\n[O][O][O]\n[O][D][D]"	P
	2 Trip has less than 13 passengers	passenger[i].filled = 1, where i is 0 - 7	prints"[X][X][X]\n[X][X][X]\n[X][X][O]\n[O][O][O]\n[O][D][D]"	prints"[X][X][X]\n[X][X][X]\n[X][X][O]\n[O][O][O]\n[O][D][D]"	P
	3 Trip is at full capacity	passenger[i].filled = 1, where i is 0 - 15	prints "[X][X][X][X]\n[X][X][X][X]\n[X][X][X][X]\n [X][X][X]\n[X] [D][D]"	prints "[X][X][X][X]\n[X][X][X][X]\n[X][X][X][X]\n [X][X][X]\n[X] [D][D]"	P
searchPassenger	1 Last name is present in the embarkation cards	lastname = "Brunner", trips contain 1 person with the last name "Brunner"	prints"[NOTIFICATION] 1 people with the lastname "Brunner" found."	prints"[NOTIFICATION] 1 people with the lastname "Brunner" found."	P
	2 Last name is not present in the embarkation cards	lastname = "Brunner", trips contains no one with the last name "Brunner"	prints"[NOTIFICATION] 0 people with the lastname "Brunner" found."	prints"[NOTIFICATION] 0 people with the lastname "Brunner" found."	P
	3 More than 1 instance of the last name is present	lastname = "Brunner", trips contains 3 people with the last name "Brunner"	prints"[NOTIFICATION] 3 people with the lastname "Brunner" found."	prints"[NOTIFICATION] 3 people with the lastname "Brunner" found."	P
inputDate	1 All inputs are valid	date.nDay = 14, date.nMonth = 6, date. nYear = 2022	returns date where date.nDay = 14, date.nMonth = 6, date.nYear = 2022	returns date where date.nDay = 14, date.nMonth = 6, date.nYear = 2022	P
	2 one of the inputs are invalid	date.nDay = 14, date.nMonth = 13, date. nYear = 2022	prints"[INPUT ERROR] Invalid Input. Allowed inputs are integers 1 to 12." and asks user for another input	prints"[INPUT ERROR] Invalid Input. Allowed inputs are integers 1 to 12." and asks user for another input	P
saveFile	1 Trips contains embarkation cards	date = 17 06 2022 (dd mm yyyy) and some trips have contents	prints "[SUCCESS] All inputs have been encoded in **Trip-17-06-2022.txt**" and creates a txt file	prints "[SUCCESS] All inputs have been encoded in **Trip-17-06-2022.txt**" and creates a txt file	P
	2 Trips are empty	date = 17 06 2022 (dd mm yyyy) and none of the trips have contents	prints "[WARNING] There are currently no inputs, so no new file was created"	prints "[WARNING] There are currently no inputs, so no new file was created"	P
loadCard	1 Text file is complete	All inputs are valid and are in the specified format	prints the Embarkation card and assigns the Embarkation card to the specified trip	prints the Embarkation card and assigns the Embarkation card to the specified trip	P
	2 Text file contains an incomplete input	last line is blank	asks the user for a drop-off point input then prints embarkation card when the input is valid	asks the user for a drop-off point input then prints embarkation card when the input is valid	P
	3 Text file contains an invalid input	embarkat point = 1 and trip number = AE150	prints "[WARNING] Embarkation Point changed from 1 to 2. (This is due to your embarkation point not matching your Trip Number" and embarkation point = 2	prints "[WARNING] Embarkation Point changed from 1 to 2. (This is due to your embarkation point not matching your Trip Number" and embarkation point = 2	P

	4	Text file does not exist	user input "x.txt" but there is no file named "x.txt	prints "[INPUT ERROR] x.txt does not exist, please try again." and returns to the main function	prints "[INPUT ERROR] x.txt does not exist, please try again." and returns to the main function	P
loadTrip	1	Trip file exists	"Trip-16-06-2022.txt" exists	Loads all of the inputs into trips[]	Loads all of the inputs into trips[]	P
	2	Trip file does not exist	"Trip-14-06-2022.txt" does not exist	prints "[INPUT ERROR] Trip-14-06-2022.txt does not exist. If this is your first time running today, please ignore this message." and returns to main	prints "[INPUT ERROR] Trip-14-06-2022.txt does not exist. If this is your first time running today, please ignore this message." and returns to main	P