Arrows Express Line Embarkation System

Donnald Miguel Robles' Machine Project Test Cases

int tripStringToNumber(char *strTripString) - Converts a trip string to its integer equivalent

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A Manila to Laguna Shuttle	strTripString = "AE101"	returns 1	returns 1	P
2	A Laguna to Manila Shuttle	strTripString = "AE150"	returns 10	returns 10	P
3	A Shuttle String Outside of Range	strTripString = "AE161"	returns 0	returns 0	P

int dropoffStringToNumber(char *strDropoffString) - Converts a dropoff string to its integer equivalent

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A string with first letter being integers and in dropoff range	strDropoffString = "1st drop-off point - Mamplasan Toll Exit"	returns 1	returns 1	P
2	A string with first letter being integers but not in dropoff range	strDropoffString = "100th drop-off point"	returns 100	returns 100	P
3	A string without first letters being ints	strDropoffString = "Sample String"	returns 0	returns 0	P

int normalizeTripString(int nTripNum) - Converts array form trip number to normalized version

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A Manila to Laguna Trip Number	nTripNum = 9	returns 109	returns 109	P
2	A Laguna to Manila Trip Number	nTripNum = 20	returns 160	returns 160	P
3	A number out of bounds	nTripNum = 21	Outputs "Invalid Trip Number" before getting passed to function prompts user to input again	Outputs "Invalid Trip Number" before getting passed to function prompted user to input again	P

char *getDropoffString(int nDropOff, int nTripNum) - Gets the shuttle's full dropoff text

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A valid Trip Number and Dropoff Number	nTripNum = 1 nDropoff = 1	1st drop-off point - Mamplasan Toll Exit	1st drop-off point - Mamplasan Toll Exit	P
2	A invalid Trip Number and valid Dropoff Number	nTripNum = 21 nDropoff = 1	Outputs "Invalid Trip Number" before getting passed to function prompts user to input again	Outputs "Invalid Trip Number" before getting passed to function prompted user to input again	P
3	A valid Trip Number and invalid Dropoff Number	nTripNum = 1 nDropoff = 10	Outputs "Invalid Dropoff" before getting passed to function prompts user to input again	Outputs "Invalid Dropoff" before getting passed to function prompted user to input again	P

int checkID(int nID) - checks if the given ID is a valid DLSU ID

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A valid DLSU ID	nID = 12146153	returns 0	returns 0	P
2	An invalid DLSU ID	nID = 12112112	returns an integer not 0	returns 4	P
3	A random integer not in DLSU ID Format	nID = 1	returns an integer not 0	returns 1	P

int checkDropoff(int nTripNum, int nDropOff) - checks if the given dropoff point is valid, returns 0 if valid

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A valid Trip Number and Dropoff Number	nTripNum = 1 nDropoff = 1	returns 0	returns 0	P
2	A invalid Trip Number and valid Dropoff Number	nTripNum = 21 nDropoff = 1	returns 1	returns 1	P
3	A valid Trip Number and invalid Dropoff Number	nTripNum = 1 nDropoff = 10	returns 1	returns 1	P

int getDropCount(struct passenger passengers[16], int nGivenNumber) - counts the number of passengers using a certain dropoff point

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	A valid given number (1 - (2-5) depending on passenger trip)	nGivenNumber = 1	Count of passengers with dropoff point 1	returns count of passengers with dropoff point 1	P
2	A positive integer outside of range	nGivenNumber = 6	returns 0 (nobody will have that dropoff point)	returns 0	P
3	A negative integer outside of range	nGivenNumber = -1	returns 0 (nobody will have that dropoff point)	returns 0	P

void findShuttleSeat(struct shuttle shuttles[20], struct passenger applicant, int nTripNum) - finds a seat for applicant

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	Applicant chosen trip has a free seat	shuttles[tripNum - 1].count = 10	Applicant gets the next seat	Applicant gets seat 11	P
2	Applicant chosen trip is full. Applicant does not have a higher priority group than anyone in trip	applicant.nPriorityGroup = 6 nTripNum = 1 passenger.nPriorityGroups = {1, 2, 3, 3, 2, 4, 5, 5, 1, 3, 4, 4, 5, 1, 5, 4}	The applicant will not be given a seat	The applicant was not given a seat	P
3	Applicant chosen trip is full. Applicant is in a higher priority group than someone in trip There are other trips going to chosen route.	applicant.nPriorityGroup = 1 nTripNum = 1 passenger.nPriorityGroups = {1, 2, 3, 3, 2, 4, 5, 6, 1, 3, 4, 4, 5, 1, 6, 4}	Latest passenger with lowest priority gets moved and applicant gets their seat	Passenger in seat gets moved Applicant gets seat 15	P
4	Applicant chosen trip is full. Applicant is in a higher priority group than someone in trip There are no other trips going to chosen route.	applicant.nPriorityGroup = 1 nTripNum = 1 passenger.nPriorityGroups = {1, 2, 3, 3, 2, 4, 5, 6, 1, 3, 4, 4, 5, 1, 6, 4}	Latest passenger with lowest priority gets moved to "special shuttle" and applicant gets their seat	Passenger in seat gets moved to "special shuttle" Applicant gets seat 15	P

void updateFile(char *strFileName, struct shuttle shuttles[20]) - updates the file with information from shuttles array

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	All contents in shuttle array are valid	Shuttle array with valid content	Outputs all shuttle contents formatted into file	Outputs all shuttle contents formatted into file	P
2	Some contents in shuttle array are valid, some are invalid	Shuttle array with some contents being valid, some being invalid	Outputs all valid shuttle contents formatted into file	Outputs all valid shuttle contents formatted into file	P
3	All contents in shuttle array are invalid	Shuttle array with invalid content	Outputs nothing into the file (the file becomes empty)	Outputs nothing into the file (the file became empty)	P

void readFile(struct shuttle *shuttles, char *strFileName) - reads a file and replaces shuttle array contents with file contents

#	Description	Sample Input Data	Expected Output	Actual Output	P/F
1	File has valid contents	File with valid contents	Shuttle array gets filled properly	Shuttle array gets filled properly	P
2	File does not exist	No file	File will be created Shuttle array will be empty	File is created Shuttle array is empty	P
3	File is empty	A blank file	Shuttle array will be empty	Shuttle array is empty	P