**Assignment:**

1. The dealer’s cost of a car is 85% of the listed price. The dealer would accept any offer that is at least $500 over the dealer’s cost. Design an algorithm that prompts the user to input the list price of the car and print the least amount that the dealer would accept for the car.
2. You are given a list of students’ names and their test scores. Design an algorithm that does the following:
3. Calculates the average test score.
4. Determines and prints the names of all the students whose test scores are below the average test score.
5. Determines the highest test score.
6. Prints the names of all the students whose test scores are the same as the highest test score.

(Each of the parts a, b, c, and d must be solved as a subproblem. The main algorithm combines the solutions of the subproblems.)

# Answers:

## Problem #1:

### Algorithm Steps

1. GET the **listPrice** of the car (provided from dealer)
2. With the **listPrice**, calculate the **dealerCost** of the car using the following formula
   1. **dealerCost** = **0.85 \*** **listPrice**
3. GET the **offer** from the customer.
4. Determine if the offer is acceptable using the following formula:

IF (offer is greater than or equal to dealerCost + $500)

THEN it is an acceptable offer. Accept Offer.

THEN Move to Step 5.

ELSE, the offer is NOT an acceptable offer.

THEN reject offer.

IF (the dealer gets a new offer from the customer)

THEN return to Step 3.

ELSE Move to step 5.

1. End transaction with customer

## Problem #2:

### Algorithm Steps

#### a) Calculate average test score

1. Sum all the **studentScores** provided from the list to calculate the total test score
   1. **totalTestScore = (studentScore[1] + studentScore[2] + … studentScore[i])**
2. Determine the average test score by using the following formula:
   1. **avgTestScore** = **totalTestScores / total number of students**

#### b) Print names of students with scores below avgTestScore

Once the average test score has been determined, we can use that to determine who is below the average and print their names by using the following logic:

IF (studentScore[1] is less than the avgTestScore)

THEN print studentName[1]

OTHERWISE

IF (studentScore[2] is less than the avgTestScore)

THEN print studentName[2]

OTHERWISE

IF (studentScore[i] is less than the avgTestScore)

THEN print studentName[i]

//repeat for all studentScores in provided score list

#### c) Determine highest test score

To efficiently determine the **highestTestScore**, each score must be compared to the next score on the list of scores provided by using the following logic:

IF (studentScore[i] is greater than or equal to studentScore[i+1])

THEN highestTestScore = studentScore[i]

OTHERWISE

highestTestScore = studentScore[i+1]

IF (highestTestScore is greater than or equal to studentScore[i+2])

THEN the highestTestScore does not change

OTHERWISE

highestTestScore = studentScore[i+2]

//repeat the above steps until there are no more students to compare. The last amount that is determined for **highestTestScore** is the highest test score achieved

#### d) Print names of students with scores that are the same as the highest test Score

Similar to printing the names of students with scores below the average test score, once the **highestTestScore** has been determined, we can use that to determine which students have the same score as the highest score and print their names by using the following logic:

IF (studentScore[1] is equal to highestTestScore)

THEN print studentName[1]

OTHERWISE

IF (studentScore[2] is equal to highestTestScore)

THEN print studentName[2]

OTHERWISE

IF (studentScore[i] is equal to highestTestScore)

THEN print studentName[i]

//repeat for all studentScores in provide score list