Problem 1

INPUT	PROCESS	OUTPUT
- 6 variables to hold quantity, price, extendedPrice, discountedPrice, discountedAmount, and totalDiscountedPrices - Asks and stores input for quantity and price of item	- While loop to continue the code until the user presses ctrl + z - Multiplies quantity and price to get extendedPrice - Sets discountedAmount to 0 before using the variable - If quantity is above 1000, multiply extendedPrice by 0.1 to get discountedAmount - Subtracts discountedAmount to extendedPrice to get discountedPrice - Adds discountedPrice to totalDiscountedPrices to get totalDiscountedPrices	- Displays the quantity - Displays the price - Displays the extended price - Displays the discounted amount - Displays the total discounted prices

Problem 2

INPUT	PROCESS	OUTPUT
- 5 variables to hold lastName, hits,	- While loop to continue the code	- -

atBats, playerCount, and battingAverage - Asks and stores input for lastName of player, number of hits, and number of at bats	until the user presses ctrl + z - If atBats is greater than 0, multiply hits by 1 and then divide the product by atBats to get battingAverage - Adds 1 count to playerCount to count total players - If any number is less than or equal to 0, return –1 (error code	- - Displays player - Displays batting average - Displays number of players

Problem 3

INPUT	PROCESS	OUTPUT
- 6 variables to hold city, tripCount, miles, gallons, milesPerGallon, and totalMiles - Asks and stores input for city name, miles traveled, and gallons used	- While loop to continue the code until the user presses ctrl + z - Divies miles by gallons to get milesPerGallon if gallons is greater than 0 - Adds miles to totalMiles to get totalMiles - Adds 1 count to tripCount to get the total number of trips - Returns –1 (error code) if gallons is	- Displays city - Displays miles per gallon - Displays total miles travelled - Displays number of trips

	less than or equal to 0	

Problem 4

INPUT	PROCESS	OUTPUT
- 5 variables to hold lastName, jobCode, hoursWorked, employeeCount, payRate, pay, averagePay, and totalPay - Asks and stores input for last name, job code, and hours worked	- While loop to continue the code until the user presses ctrl + z - Set payRate to $25 if jobCode is equal to "L" - Set payRate to $30 if jobCode is equal to "A" - Set payRate to $50 if jobCode is equal to "J" - Returns –1 (error code) if jobCode is not equal to L, A, or J - if hours worked is greater than 40, calculate the ovetime pay - Multiplies payRate by hoursWorked to get pay if hours worked is 40 or less - Adds pay to totalPay to get totalPay (overall)	- Displays last name - Displays job code - Displays hours worked - Displays pay amount - Displays total pay for all employees - Displays number of employees - Displays average pay

	- Add 1 count to employeeCount to find total number of employees	

Problem 5

INPUT	PROCESS	OUTPUT
- 8 variables to hold lastName, districtCode, creditHours, studentCount, totalCreditHours, tuitionRate, tuitionOwed, and totalTuition - Asks and stores input for last name, credit hours, and district code	- While loop to continue the code until the user presses ctrl + z - If atBats is greater than 0, multiply hits by 1 and then divide the product by atBats to get battingAverage - Sets tutionRate to $250 if districtCode is equal to "I" - Sets tutionRate to $550 if districtCode is equal to "O" - Returns error code if disctrict code does not equal to "I" or "O" - Multiplies tuitionRate by creditHours to get tuitionOwed - Adds tuitionOwed to totalTuition to get totalTuition(overall)	- Displays student - Displays tuition owed - Displays total tuition owed - Displays total credit hours - Displays total number of students

	- Adds creditHours, to totalCreditHours to get totalCreditHours (overall) - Adds 1 to studentCount to get total number of students	