Problem Set – While Loops. Develop IPO for each of the problems below and then save within this document. Then Write the code files using C++. Upload both IPO and .CPP files to Blackboard.

1. Allow any number of users to enter a quantity and price at the keyboard (use ctl+z to stop). Compute the extended price (quantity times price). If quantity is over 1000 give a 10% discount. Display quantity, price and extended, discount amount and discounted price for each entry. Keep a sum of the total for all the discounted prices. Display the total of discounted prices after all entries have been entered.

input	process	output
Quantity,price	extended = quantity × price if quantity > 1000 → discount = 10% discounted = extended – discount add discounted to running total	quantity, price, extended, discount, discounted price, total discounted prices

2. Allow any number of players to enter last name, number of hits and at bats at the keyboard (use ctl+z to stop). Compute batting average (hits/ at bats). Display last name and batting average for each player. Keep a count of the number of players (or entries) made. Display the count after all entries have been made.

input	prcoess	output
last name, hits, at-bats	batting average = hits ÷ at-bats; increment count of players	last name, batting average, total number of players

3. Enter destination city, miles travelled to get there and gallons of gasoline used for any number of trips entered at the keyboard (use ctl+z to stop). Compute miles per gallon (miles travelled / gallons used). Display the destination city and miles per gallon for each trip entered. Sum the miles travelled and give a count of the number of trips made. Display these at the end of the program.

input	process	output
destination, miles, gallons	mpg = miles ÷ gallons; accumulate total miles; count trips	destination, mpg, total miles, number of trips

4. Allow the employee to enter last name, job code and hours worked (use ctl+z to stop). Calculate pay. (Job code L is $25/hr, A is $30/hr and J is $50/hr). Give time and a half for overtime. Display last name, job code, hours worked and pay for each employee. Sum the pay for each employee as well as count the entries made. After all entries are made, compute and display the average pay and the number of entries made.

input	process	output
last name, job code, hours	set pay rate (L=$25, A=$30, J=$50) if hours > 40 → overtime at 1.5 × rate accumulate total pay, count employees	employee info, pay; average pay, number of employees

5. Allow the user to enter student last name, credit hours and district code for any number of students (use ctl+z to stop). Compute tuition owed. In district (code of I) is charged $250 per credit hour. Out of district (code of O) is $550 per credit hour. Display student name and tuition

owed for each entry. Sum the amount of tuition owed for all students as well and the total credit hours taken and finally the number of students who entered data. Display total tuition, total credit hours taken and count of number of students at the end.

input	process	output
last name, credit hours, district code	if district = I → $250/hr, if O → $550/hr tuition = hours × rate accumulate total tuition, total credits, count students	student info, tuition owed; totals for all students