Problem Set – Pass by Reference Functions. Create an IPO for each problem within this document. Save the document and upload to Blackboard for grading. Then create a C++ program for each problem. Upload the .CPP files to Blackboard for grading.

1. Allow the user to enter a quantity and price, use ctl+z to stop. Use one function to compute the total (quantity times price), tax (7% of total) and total order (total plus tax). The function should be passed the quantity and price by value and total, tax and total order by reference. Display total, tax and total order in main. Sum and display total of all orders and tax for all orders and display after the loop (all data is processed).

Input	process	output

Quantity, price	total = quantity × price
	tax = total × 0.07
	total_order = total + tax
		total, tax, total_order

(quantity and price passed by value) (total, tax, total_order passed by reference)		Sum of all totals and taxes after loop

2. Enter the weight of a package and zip code. Use ctl+z to stop. Use a single function to do the computations specified next. Pass these weight and zip code by value, and pass postage, area charge and weight charge by reference. Compute postage to be sum of weight charge and area charge. Use tables below to find the charges. Compute weight charge to be weight x weight charge per ounce. Find the area charge in the table based on zip code. Then compute postage to be area charge plus weight charge. The function should return the weight charge, area charge and postage. Display area charge, weight charge and postage. Count and display the number of entries made.

Area Table – Used to determine the area charge

Area	Area Charge
60171	$2.00
60172	$2.50
60635	$3.00

All others	$5.00

Weight Table – used to determine the weight charge

Weight	Weight Charge per Ounce
>100	0.02
>50	0.03
All other	0.05

Input	process	output
weight, zip_code	Determine area_charge based on zip code: 60171 = 2.00 60172 = 2.50 60635 = 3.00 All others = 5.00	area_charge, weight_charge, postage Count/display total number of entries
	Determine weight_charge_per_ounce: Greater than 100 = 0.02 Greater than 50 = 0.03 All others = 0.05	

	weight_charge = weight × weight_charge_per_ounce
	postage = area_charge + weight_charge	

3. Enter the student's last name, credit hours and financial aid, use ctl+z to stop. Pass credit hours and financial aid to a function by value. Pass tuition and tuition owed by reference. Compute tuition to be credit hours times $250. Compute tuition owed to be tuition minus the financial aid. Display student's last name, tuition and tuition owed. Sum and display total tuition owed by all students, count of number of entries and average amount owed by students.

Input	process	output

last_name, credit_hours, financial_aid	tuition = credit_hours × 250
	tuition_owed = tuition − financial_aid
		last_name, tuition, tuition_owed
		Display total tuition owed, number of students, and average owed

4. Enter a number of widgets, use ctl+z to stop. Pass the number to a function by value, use ctl+z to stop. Use a single function to determine the cost per widget using the cost table below. Then compute extended price (number of widgets x cost per widget) and 7% sales tax. Finally compute total order to be extended price plus sales tax. Pass cost per widget, extended price, sales tax and total order by reference. For each line, display number of widgets, cost per widget, extended price, sales tax and total order. Sum all total orders and display when there is no more data to process.

   Cost Table

Number of Widgets	Cost Per Widget
10000 and up	4.00
5,000 and up	5.00
All other amounts	10.00

Input	process	output
num_widgets	Determine cost_per_widget: 10000 and up = 4.00 5000 and up = 5.00 All other amounts = 10.00	

	extended_price = num_widgets × cost_per_widget
	sales_tax = extended_price × 0.07
	total_order = extended_price + sales_tax
		num_widgets
		cost_per_widget
		extended_price
		sales_tax
		total_order
		sum of all total orders

5. Enter the amount of investment, the 5 year interest rate and 10 year interest rate, use ctl+z to stop. Pass the amount and interest rates to a function by value. Pass variables representing five year amount and ten year amount to the same function by reference. Compute the five year amount and ten year amount using the formula below. Display the amount of the investment, the five year amount and the ten year amount.

   Five year amount = amount of the investment x (1 + 5 year rate) raised to 5th power.

   Ten year amount = amount of the investment x (1+10 year rate) raised to 10th power.

   Note: Enter the 5 year rate and 10 year rate in decimal form, i,e 5% is entered as 0.05.

   Also Note: you need to use the pow built in function. Recall the pow function syntax:

   pow (base, exponent)

   In this case, base is 1 + 5 year rate and exponent is 5.

   Another line will be: base is 1 + 10 year rate and exponent is 10.

   Use #include<math.h> for the pow function.

Input	process	output

amount, five_year_rate, ten_year_rate	five_year_amount = amount × pow((1 + five_year_rate), 5)
	ten_year_amount = amount × pow((1 + ten_year_rate), 10)
		amount, five_year_amount, ten_year_amount