Problem Set – Introduction to Functions - Victoria Plowiec

1. Allow the user to repeatedly enter a quantity and price. Prompt the user on whether they want to do the program (Yes or No). Use a function to compute the total (quantity times price). The function should be passed the quantity and price and then return the total. In the function, provide a 10% discount if the total is over $10,0000.00. Display quantity, price and total. Sum and display the extended price.

Input	**Process**	**Output**
Ask the user if they want to run the program (Yes or No) Quantity (number of items being bought) Price (cost per item)	Call a function and send quantity and price Inside the function: total = quantity * price If total > 10000, apply discount: discount = total * 0.10 total = total – discount Return total Display quantity, price, and total Add totals together: grand_total = grand_total + total Repeat if user enters Yes	Show quantity, price, total Show grand total

2. Enter players last name, number of hits and at bats at the keyboard. Prompt the user on whether they want to do the program (Yes or No). Use a function to compute batting average. Pass the hits and at bats to the function. The function should return batting average. Display last name and batting average. Give a count of the number of players entered.

Input	**Process**	**Output**
Ask the user if they want to enter a player (Yes or No) Player last name Hits (number of successful hits)	Send hits and at-bats to function Inside the function: batting_average = hits / at_bats	Show player last name Show batting average At the end, show how many players were entered

At-bats (number of times the player batted)	Return batting_average Display last name + batting average Add to player counter: player_count = player_count + 1 Repeat if user enters Yes	

3. Enter the destination city, miles travelled and gallons used for a trip. Prompt the user on whether they want to do the program (Yes or No). Use a function to compute miles per gallon and cost of gas. Pass miles travelled and gallons used to the function. The function should return miles per gallon anf compute gas cost to be gallons times 3.00. Count the number of entries made (number of trips) Display destination city, miles, mpg and gas cost. At end display the number of entries made, total miles travelled for all trips and total gas cost of all trips.

Input	**Process**	**Output**
Ask the user if they want to enter a trip (Yes or No) Destination city Miles traveled Gallons of gas used	Send miles and gallons to function Inside the function: mpg = miles / gallons gas_cost = gallons * 3.00 Return mpg and gas_cost Display city, miles, mpg, and gas cost Add trip count: trip_count = trip_count + 1 Add total miles: total_miles = total_miles + miles Add total gas cost: total_gas_cost = total_gas_cost + gas_cost	Show destination city for each trip Show miles, mpg, and gas cost for each trip At the end, show: Number of trips Total miles traveled Total gas cost for all trips

	Repeat if user enters Yes	

4. Allow the employee to enter last name, job code and hours worked. Prompt the user on whether they want to do the program (Yes or No). Use a function to determine the pay rate. Pass to this function the job code and it should return rate of pay and gross pay. Use Job code L is $25/hr, A is $30/hr and J is $50/hr for respective pay rates. Compute gross pay. Give time and a half for overtime. Display last name,hours, pay rate and gross pay. Sum and display total of all gross pay.

Input	**Process**	**Output**
Ask the user if they want to enter an employee (Yes or No) Employee last name Job code (L, A, or J) Hours worked	Send job code to function Inside the function: If job code = L → rate = 25 If job code = A → rate = 30 If job code = J → rate = 50 Return rate In main program: If hours > 40 (overtime): overtime_hours = hours – 40 overtime_pay = overtime_hours * (rate * 1.5) regular_pay = 40 * rate gross_pay = regular_pay + overtime_pay Else (no overtime): gross_pay = hours * rate Display last name, hours, rate, gross pay Add to total gross pay:	Show employee last name Show hours Show pay rate Show gross pay At the end, show total gross pay for all employees

	total_gross = total_gross + gross_pay Repeat if user enters Yes	

5. Allow the user to enter student last name, credit hours and district code. Prompt the user on whether they want to do the program (Yes or No). Use a function to compute tuition owed. Charge In district (code of I) $250 per credit hour. Out of district (code of O) is $550 per credit hour. The function should receive credit hours and district code and return tuition owed. Display student name and tuition owed. Sum and display total of all tuition owed.

Input	**Process**	**Output**
Ask the user if they want to enter a student (Yes or No) Student last name Credit hours District code (I for in-district, O for out-of-district)	Send credit hours and district code to function Inside the function: If district code = I: tuition = credit_hours * 250 If district code = O: tuition = credit_hours * 550 Return tuition Display student last name and tuition owed Add to total tuition: total_tuition = total_tuition + tuition Repeat if user enters Yes	Show student last name Show tuition owed At the end, show the total tuition for all students