Problem Set – Introduction to Functions.

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

qty	comp_ext_price(qty, unit_price)
	    ext_price = qty * unit_price
	
	    if ext_price > 100000.00
	        disc_amt = ext_price * 0.10
	    else:
	        disc_amt = 0
	
	    disc_ext_price = ext_price – disc_amt
	
	    return disc_ext_price
		qty

price	Main
	    total_ext_price = 0
	
	    Do you want to start this program (Yes or No)?
	
	    While (Yes)
	        Input qty and price
	        ext_price = comp_ext_price(qty, unit_price)
	        display ext_price
	        display qty
	        total_ext_price = total_ext_price + ext_price
	        Do you want to continue with this program?
		ext_price

	Display total_ext_price	total
		unit_price

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

last_name	comp_batting_avg(num_of_hits,num_of_bats)
	        batting_avg = num_of_hits / num_of_bats
	        return batting_avg
		last_name

num_of_hits	Main num_of_players = 0 Do you want to start this program (Yes or No)? While(Yes) Input last_name, num_of_hits, and num_of_bats batting_avg = comp_batting_avg(num_of_hits,num_of_bats) display last_name and batting_avg num_of_players = num_of_players + 1 Do you want to continue this program?	
num_of_bats	Display num_of_players	batting_avg
		num_of_players

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 and 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

dest_city	comp_mpg_cost (miles_travelled,gallons_used)
	mpg = miles_travelled / gallons_used
	cost_of_gas = gallons_used * 3.00
	return mpg, cost_of_gas
		dest_city

miles_travelled	Main
	num_of_trips = 0
	total_miles = 0
	total_gas_cost = 0
	Do you want to start this program (Yes or No)?
	While(Yes)
	Input dest_city, miles_travelled, and gallons_used
	mpg = comp_mpg_cost(miles_travelled,gallons_used)
	cost_of_gas = comp_mpg_cost(miles_travelled,gallons_used)
	Display dest_city, miles_travelled, mpg, and cost_of_gas
		miles_travelled

	num_of_trips = num_of_trips + 1
	total_miles = total_miles + miles_travelled
	total_gas_cost = total_gas_cost + cost_of_gas
	Do you want to continue the program?	

gallons_used	Display num_of_trips, total_miles, total_gas_cost	mpg
		cost_of_gas
		num_of_trips
		total_miles
		total_gas_cost

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

last_name	comp_pay_rate(job_code,hrs_worked)
	if job_code == ‘L’:
	pay_rate = 25
	elif job_code == ‘A’:
	pay_rate = 30
	elif job_code == ‘J’:
	pay_rate = 50
	
	if hrs_worked > 40:
	overtime_pay = (pay_rate * 1.5)*(hrs_worked – 40)
	regular_pay = pay_rate * 40
	gross_pay = overtime_pay + regular_pay
	else:
	gross_pay = pay_rate * hrs_worked
	
	return pay_rate, gross_pay
		last_name

job_code	Main
	total_gross_pay = 0
	
	Do you want to start the program (Yes or No)?
	
	while(yes)
		hrs_worked

	Input last_name, job_code, hrs_worked
	pay_rate = comp_pay_rate(job_code,hrs_worked)
	gross_pay = comp_pay_rate(job_code,hrs_worked)
	Display last_name, hrs_worked, pay_rate, and gross_pay
	total_gross_pay = total_gross_pay + gross_pay
	Do you want to continue the program (Yes or No)?	

hrs_worked	total_gross_pay = total_gross_pay + gross_pay	pay_rate
	Display total_gross_pay	gross_pay
		total_gross_pay

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

last_name	comp_tuition_owed(credit_hrs,district_code)
	if district_code == ‘I’:
	cost_credit = 250
	elif district_code == ‘O’:
	cost_credit = 550
	tuition_owed = cost_credit * credit_hrs
	return tuition_owed
		last_name

credit_hrs	Main
	total_tuition_owed = 0
	Do you want to start the program (Yes or No)?
	while(Yes):
	Input last_name, credit_hrs, and district_code
	tuition_owed = comp_tuition_owed(credit_hrs,district_code)
	Display last_name, tuition_owed
	total_tuition_owed = total_tuition_owed + tuiton_owed
		tuition_owed

	Do you want to continue this program (Yes or No)?	
district_code	Display total_tuition_owed	total_tuition_owed