Problem Set - Functions Pass By Value

Create a loop that uses a signal (see previous lesson on loops) to stop. In all the problems user entries are done repeatedly (in a loop) until the signal.

1. Allow the user to enter quantities and prices in a loop. 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 total extended price.

Input	Process	Output
Arguments: Qty, unitprice	Function: Compute the extended price, quantity times unit price.	Return: Extended price
qty , price from user	Read user input and compute extended price	Qty, price, Extprice for each item totalExtPrice for all items

2. Enter players last name, number of hits and at bats at the keyboard. 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
Hits, at bats	Compute batting average: hits/at bats	Batting average
Last name, hits, at bats	Read user input and compute batting average	Last name, batting average Count the number of players entered

3. Enter the destination city, miles travelled and gallons used for a trip. Use a function to compute miles per gallon. Pass miles travelled and gallons used to the function. The function should return miles per gallon. Count the number of entries made (number of trips) Display destination city, miles and mpg. At end display the number of entries made.

Input	Process	Output
Total miles, gallons	Compute miles per gallon: miles / gallons	Miles per gallon
City, total miles, gallons	Read user input then Compute miles per gallon	City, miles, miles per gallon. # of trips made

4. Allow the employee to enter last name, job code and hours worked. Use a function to determine the pay rate. Pass to this function the job code and it should return rate of 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 and gross pay. Sum and display total of all gross pay.

Input	Process	Output
Job code, hours	Job code * hours If Job code == L, = 25 If Job code == A, = 30 If Job code == J, = 50 If hours > 40, (1.5 * job code * hours)	Gross pay
Last name, job code, hours	Read user input and compute gross pay.	Last name, gross pay Sum total gross pay

5. Allow the user to enter student last name, credit hours and district code. 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
Credit hours, district code	Credit hours * district code If district code == I, = 250 If district code == O, = 550	tuition
Last name, credit hours, district code	Read user input and compute tuition owed	Last Name, tuition Sum total of all tuitions owed

Examples

1. Enter the number of Points and redemption code. For redemption code C then compute value as 2 x rewards points. Redemption code X then they get 3 x rewards points. All other codes get 1.5 x rewards points. Write a function that receives points and redemption code and computes rewards points. Display points, redemption code and rewards points.
2. Enter two numbers and operation code (A, S, M, D). Write a function that receives the two numbers and uses the operation code to perform an operation on the two numbers (A=addition, S=Subtraction, M=Multiplication, D=Division). Check for dividing by 0. If the second number is 0 then set result to -999. Display two number, operation code, result and message if attempt to divide by zero.
3. Allow the user to enter a string. The string can be entered with any case (all upper, all lower of mixed). Write a function that accepts the string and returns all lower case when the original string is all upper or mixed. If the original string is all lower then make the string all upper case. The function should return the new string. Display both the original and new string.