|  |
| --- |
|  |
| **Programming for Analytics – CA One** |
|  |
|  |
| **Nagarathna Sali** |
|  |

|  |
| --- |
|  |

**Problem 1**

Write a program in Python to prompt the user to input their name, employee number, week ending date, hours worked, rate per hour, standard and overtime tax percentage rate. Use the data input to calculate gross pay, tax deductions and net pay. Output the results as a formatted payslip. Assume that a standard working week is 37.5 hours.

E.g. Ask the user to enter the following data:

Employee Name: (sample input – Mark Bate)

Employee Number: (sample input – 123456789A)

Week ending: (sample input - 26/01/2018)

Number of hours worked: (sample input – 42.5)

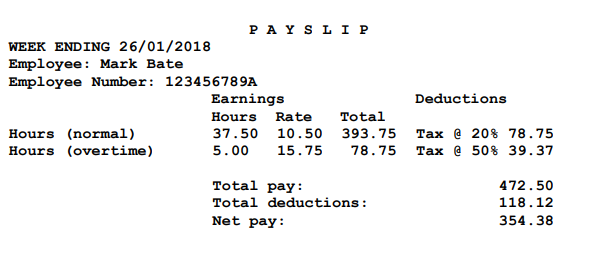
Hourly Rate: (sample input – 10.50)

Overtime Rate: (time-and-a-half as 1.5)

Standard Tax Rate: (sample input – 20)

Overtime Tax Rate: (sample input – 50)

Once the above data has been entered the program should display the employee’s payslip as per the following example:



**Problem 1 – Solution**

empName=input("Employee Name: ")

empNumber=input("Employee Number: ")

weekEnding=str(input("Week ending: ")) #input for week ending in date format

noOfHoursWorked=float(input("Number of hours worked: "))

hourlyRate=float(input("Hourly Rate: ")) #hourly rate for standard working hours

overtimeRate=float(input("Overtime Rate: ")) # rate for overtime hours

stdTaxRate=float(input("Standard Tax Rate: ")) #tax rate for standard hours

overtimeTaxRate=float(input("Overtime Tax Rate: ")) #tax rate for overtime hours

##variable for total standard Hours

tot=hourlyRate\*37.50

###variable Overtime Hours is Number of hours worked minus standard working week hours 37.5

overTimeHrs=noOfHoursWorked-37.50

#####variable for overtime pay is Hourly Rate multiplied by overtime rate for example 1.5

overTimePay=hourlyRate\*overtimeRate

### variable for total overtime is Total amount to be paid for overtime is over time hours \* Overtime pay

totOvertime=overTimeHrs\*overTimePay

stdTaxCalc=stdTaxRate/100 ###standard Tax calculation percentage

overTimetaxcalc=overtimeTaxRate/100###Overtime Tax calculations percentage

dedStdTime=round((tot\*stdTaxCalc),2) ### Tax deduction for standard/normal hours

dedOvrTime=round((totOvertime\*overTimetaxcalc),2) ### Tax deduction for Overtime hours

####Total Pay =total pay for standard hours +total pay for Overtime hours

totalPay=round((tot+totOvertime),2)

#### variable for total deduction deduction for normal hours + deduction for Overtime Hours

toDeductions=round((dedStdTime+dedOvrTime),2)

##Printing payslip

print("\t\t\t\t"," ".join("PAYSLIP"),"\t\t\t")#join function will add space between letters

print("WEEK ENDING ",weekEnding)

print("Employee: ",empName.title()) #title function capitalizes each word

print("Employee Number: ",empNumber)

print("\t\t\t Earnings\t\t Deductions")

print("\t\t\t Hours\t","Rate\t","Total")

print("Hours (normal)\t\t","37.50\t",hourlyRate,"\t",tot," Tax @",stdTaxRate,"%",dedStdTime)

print("Hours (overtime)\t",overTimeHrs,"\t",overTimePay,"\t",totOvertime,"\t","Tax @",overtimeTaxRate,"%",dedOvrTime)

print()

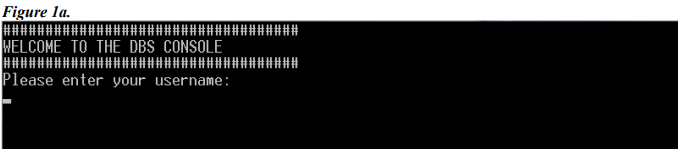
print("\t\t\t Total pay:\t\t\t ",tot+totOvertime)

print("\t\t\t Total Deductions:\t\t ",dedStdTime+dedOvrTime)

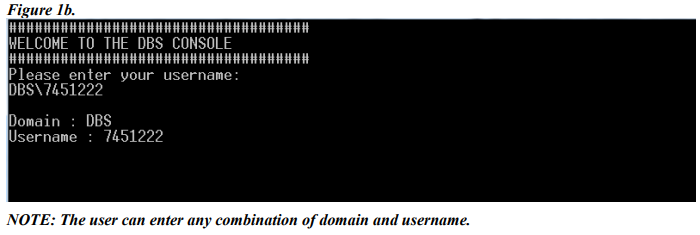
print("\t\t\t Net Pay:\t\t\t ",totalPay-toDeductions)

**Problem 2**

Write a program in Python which prompts the user for their username in the format Domain Name\Username as per Figure 1a below.



On entering their domain and username and pressing carriage return, write out to the console window each individual data item as per Figure 1b below.



**Problem 2 -Solution**

print("##################################")

print("WELCOME TO THE DBS CONSOLE")

print("##################################")

test=input("Please enter Your Username")

test1=test.split("/")#using split function to split user input based on delimiter /

str1=test1[0].upper()#saving the string before delimiter / to str1 and capitalising each letter of the string

str2=test1[1].upper()#saving the string after delimiter / to str2 and capitalising each letter of the string

if(str1=="DBS"):#if else function to find whether user entered domain first or username first

print("Domain: ",str1)

print("Username: ",str2)

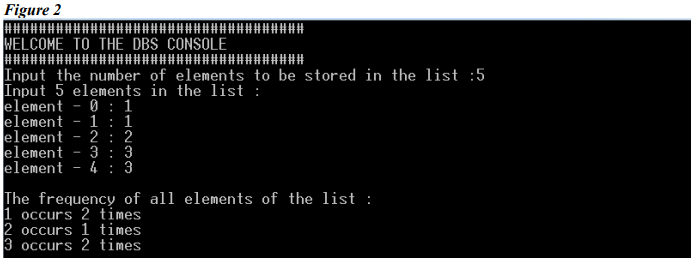
else:

print("Domain: ",str2)

print("Username: ",str1)

**Problem 3**

Write a program in Python that prompts the user to enter a number of integer values. The program stores the integers, counts the frequency of each integer and displays the frequency as per Figure 2 below.



**Problem 3 - Solution**

print("##################################")

print("WELCOME TO THE DBS CONSOLE")

print("##################################")

test=int(input("Input the number of elements to be stored in list: "))#prompt user to enter number of elements they want to store

print("Input",test,"elements in the list:")#print user input

i=0

mylist=[]#declaring empty list

import collections #importing collections to use counter function

while i<test: #since i starts from zero loop will run till i<test and will input all the integers

test1=input("element - "+str(i)+" : ")

mylist.append(test1) #appending each time to list created

i +=1 #incrementing i for iterations

my\_dict=collections.Counter(mylist)# storing thecount to dictionary

print("The frequency of all the elements of the list")

for key,value in my\_dict.items(): #loop through key ,value of dictionary by using items() function

print(key,"occurs", value,"times") #printing occurance of each number

**Problem 4**

Implement the MYPY Phone Book System in Python as per Figure 3 below which allows users to add, delete, update and lookup phone numbers. The MYPY Phone Book System should store the individual’s Full Name and Phone Number. Your program should not allow users to add the same number twice. On adding, deleting, updating or looking up a number, your program should let the user know if the operation was successful or not. On looking up a number return the full name and number of the individual; if the number is not found give the user the option to add the details they are looking up. The user can perform multiple actions; they can add a new entry and subsequently delete an entry without having to stop and start the program until they decide to quit.



**Problem 4- Solution**

####defining menu function

def mypymenu(): #defining menu for MYPY operations

print("########################")

print("MYPY PHONE BOOK")

print("########################")

print("1: Add New Entry")

print("2: Delete Entry")

print("3: Update Entry")

print("4: Lookup Numbers")

print("5: QUIT")

print()

mylist={}###intialising the dictionary

menuOpt=0 ####variable to store choice

mypymenu()

while menuOpt != 5:### this function will run until user chooses to quit

menuOpt=int(input("Choose Option: "))

if(menuOpt==1):###this will add a entry

print("Enter Full Name and Phone Number to add")

fullName = str(input("Enter Full Name: "))

phNo = int(input("Enter Phone Number: "))

if phNo in mylist:###checks whether phone number already exist else adds a record

print("Phone number already exist")

else:

mylist[phNo] = fullName # store key:value set in dictionary

print("Record Added Successfully")

elif(menuOpt==2): ###checks a phone number to delete , if phone no not found displays error

print("Removing a record")

phNo = int(input("Enter Phone number to remove: "))

if phNo in mylist:

del mylist[phNo]

print("Record deleted successfully")

else:

print(phNo, "Phone Number not found")

elif(menuOpt==3): ###checks a phone number to update , if phone no not found allows user to add a record else updates the existing record

phNo = int(input("Enter Phone Number to update"))

if phNo in mylist:

editpref = input("Want to edit Name or Number?: ")

if editpref.lower()== "name":

newfullName = input("Enter new name: ")

mylist[phNo] = newfullName

print("Update successful")

else:

newPhNo = int(input("Enter new Number: "))

newfullName = mylist[phNo]

del mylist[phNo]

mylist[newPhNo] = newfullName

print("Update successful")

else:

print("Number not found")

# phNo = int(input("Enter Phone Number: "))

# fullName = str(input("Enter Full Name: "))

#mylist[phNo] = fullName

#print("Record updated successfully")

elif (menuOpt == 4):####looks up for a phone number displays message accordingly

if mylist: #if Phonebook is not empty do below operation

print("Lookup for a phone number")

phNo = int(input("Enter Phone Number to search: "))

if phNo in mylist:

print("Name is ",mylist[phNo])

else:

print(phN

o,"Phone Number not found")

pref=input("Do you want to add new record? enter Yes or No")

if pref.lower()=="yes":

print("Enter details to Add a record")

phNo = int(input("Enter Phone Number: "))

fullName = str(input("Enter Full Name: "))

mylist[phNo] = fullName

else:

break;

else:

print("Phonebook is empty")

elif menuOpt == 5:##### this will exit from the transactions

##print("current list",mylist)

print("Exit")