SECTION A

Session 1 & 2

# Input / Output

# 1. Accept Empid,EmpName,Monthly\_Salary,Tot\_Deductions, Tot\_Allowances

# and Display Employee Name and Salary in hand

empId = int(input("Enter empId: "))

empName = input("Enter name: ")

salary = int(input("Enter salary: "))

allowances = int(input("Enter allownces: "))

print(f"EmpName: {empName}, Salary: {salary}")

# if Conditions :

# 1. Accept 3 integers from the User and Display Maximum

number1 = int(input("Enter number 1 :"))

number2 = int(input("Enter number 2 :"))

number3 = int(input("Enter number 3 :"))

if(number1>number2 and number1>number3):

    print(f"{number1} is maximum")

elif(number2>number1 and number2>number3):

    print(f"{number2} is greatest")

else:

    print(f"{number3} is greatest")

# 2. Accept 3 integers from USer and display Minimum

number1 = int(input("Enter number 1 :"))

number2 = int(input("Enter number 2 :"))

number3 = int(input("Enter number 3 :"))

if(number1<number2 and number1<number3):

    print(f"{number1} is smallest")

elif(number2<number1 and number2<number3):

    print(f"{number2} is smallest")

else:

    print(f"{number3} is smallest")

# loops (Solve without Using Functions if any)

# 1. Accept Integers from User till Users Choice and do the Following:

# 1. Sum of all Integers

# 2. Average of all Integers

# 3. Maximum Integer from all

# 4. Minimum Integer from all

list =[]

while(True):

    num = input("Enter a number, enter end to exit: ")

    if(num.upper() == 'END'):

        break

    list.append(int(num))

if(len(list)>0):

    sum = 0

    avg = 0

    max = list[0]

    min = list[0]

    for nums in list:

        sum+= nums

        if(nums>max):

            max = nums

        if(nums<min):

            min = nums

    avg = sum/len(list)

    print(f"Sum : {sum}\nAverage : {avg}\nMaximum : {max}\nMinimum : {min}")

# 2. Accept a String from User an do the following :

# 1. Find the Length

stringInput = input("Enter string: ")

lengthOfString = len(stringInput)

print(f"Length of string : {lengthOfString}")

# 2. Display String in reverse

reverseStr = ""

for i in range(len(stringInput)-1,-1,-1):

    reverseStr+= stringInput[i]

print(f"Reverse of string : {reverseStr}")

# 2. Display every alternate Character in Upper Case

for i in range(0,len(stringInput),2):

    print(f"Alternating char: {stringInput[i].upper()}")

# 3. Find out No of Vowels in the String

numberOfVowels = 0

for i in range(0,len(stringInput)):

    if(stringInput[i].lower() in ['a','e','i','o','u']):

        numberOfVowels+= 1

# 4. Accept Username and Date of Birth (dd-mon-yy) from User

userName = input("Enter userName: ")

dob = input("Enter DOB in DD-MMM-YY format: ")

print(f"UserName: {userName}")

print(f"DOB: {dob}")

# Create a Password String which will be combination of

# 1st 4 letters of username and last 2digits of Date of Birth

# followed by $ sign

password = ""

if(len(userName)<4):

    password+= userName

else:

    password+= userName[0:4]

password+= dob[-2:]

print(f"Password: {password}")

# 5. Encrypt the String and return Encrypted String

# (Assume your Algorithm)

encryptedPassword = ""

for i in range(int(len(password)/2),len(password)):

    encryptedPassword+= password[i]

for i in range(0,int(len(password)/2)):

    encryptedPassword+= password[i]

print(f"Encrypted password : {password} to {encryptedPassword}")

# 3. Write Python Program to do the following :

# 1. Display Area of

# Circle

radius = int(input("Enter radius: "))

areaOfCircle = 3.14\*radius\*radius

print(f"Area of circle with radius {radius} is : {areaOfCircle}")

# Parallelogram

base = int(input("Enter the base of parrallelogram: "))

height = int(input("Enter the height of parrallelogram: "))

areaOfParrellolagram = base\*height

print(f"Area of parrellolagram is {areaOfParrellolagram}")

# 4. Accept Integer and find Square root of Integer

toSquareNum = int(input("Enter a number to find sqaure of it"))

squared = toSquareNum\*toSquareNum

print(f"Square of {toSquareNum} is {squared}")

SECTION B 3 &4

fruits\_list = []

while True:

# Display the menu

print("\nMenu:")

print("1. Add New Fruits")

print("2. Buy Fruits")

print("3. Show Total Fruits in the List")

print("4. Exit")

option = input("Choose an option (1/2/3/4): ")

if option == '1':

# Add new fruits

n = int(input("Enter the number of fruits to add: ")) # Number of fruits to add

for i in range(n):

fruit\_name = input(f"Enter the name of fruit {i+1}: ") # Get fruit name

price = float(input(f"Enter the price of {fruit\_name} per kg: ")) # Get price

# Append price at even index, fruit name at odd index

fruits\_list.append(price)

fruits\_list.append(fruit\_name)

elif option == '2':

if len(fruits\_list) == 0:

print("No fruits available. Please add fruits first.")

else:

# Show the available fruits menu

print("\nAvailable Fruits Menu:")

for i in range(1, len(fruits\_list), 2):

print(f"{i//2 + 1}. {fruits\_list[i]} - {fruits\_list[i-1]} per kg")

total\_price = 0 # Initialize total price

# Customer selects fruits to buy

while True:

choice = input("Enter the number of the fruit to buy or 'done' to finish: ")

if choice.lower() == 'done':

break

fruit\_index = (int(choice) - 1) \* 2 # Calculate the index of the chosen fruit

total\_price += fruits\_list[fruit\_index] # Add the price to the total

print(f"\nTotal Price of Fruits Bought: {total\_price}")

elif option == '3':

if len(fruits\_list) == 0:

print("No fruits in the list.")

else:

# Show the total list of fruits and prices

print("\nTotal Fruits in the List:")

for i in range(1, len(fruits\_list), 2):

print(f"{i//2 + 1}. {fruits\_list[i]} - {fruits\_list[i-1]} per kg")

elif option == '4':

print("Thank you for using. Goodbye!")

break

else:

print("Invalid option, please try again.")

# Tuple to store employee details (EmpId, Phone Numbers)

# Each element of the tuple is a tuple itself: (EmpId, [Phone Numbers])

employees = (

(101, ["9876543210", "8765432109"]),

(102, ["9123456780"]),

(103, ["9812345678", "9900112233"]),

(104, ["9654321987"]),

(105, ["9123487650", "9988776655"])

)

# Convert tuple to list for updates

employees\_list = list(employees)

# Main program flow

while True:

print("\nMenu:")

print("1. Display Employee Phone Numbers")

print("2. Update Employee Phone Number")

print("3. Exit")

option = input("Choose an option (1/2/3): ")

if option == '1':

# Display phone numbers of the employee

emp\_id = int(input("Enter the Employee ID to search: "))

# Search for employee by ID

emp\_found = False

for emp in employees:

if emp[0] == emp\_id:

print(f"Phone numbers for Employee ID {emp\_id}: {', '.join(emp[1])}")

emp\_found = True

break

if not emp\_found:

print(f"Employee with ID {emp\_id} not found in the database.")

elif option == '2':

# Update phone number of an employee

emp\_id = int(input("Enter the Employee ID to update: "))

# Search for employee by ID

emp\_found = False

for emp in employees\_list:

if emp[0] == emp\_id:

new\_phone\_number = input("Enter the new phone number to add: ")

# Update the phone number

emp[1].append(new\_phone\_number)

print(f"Phone number {new\_phone\_number} added for Employee ID {emp\_id}.")

emp\_found = True

break

if not emp\_found:

print(f"Employee with ID {emp\_id} not found in the database.")

elif option == '3':

print("Thank you for using. Goodbye!")

break

else:

print("Invalid option, please try again.")

# Create sets for the fruits each salesman has

Fruit\_Salesman1 = {"Apple", "Banana", "Mango", "Orange", "Grapes"}

Fruit\_Salesman2 = {"Pineapple", "Banana", "Mango", "Peach", "Grapes"}

# Menu for operations

while True:

print("\nMenu:")

print("1. Find out Common Fruits with both Salesmen")

print("2. List Extra Fruits with Both Salesmen")

print("3. List Total Fruits with both Salesmen")

print("4. Exit")

option = input("Choose an option (1/2/3/4): ")

if option == '1':

# Find common fruits between both salesmen

common\_fruits = Fruit\_Salesman1.intersection(Fruit\_Salesman2)

if common\_fruits:

print(f"Common fruits between both salesmen: {', '.join(common\_fruits)}")

else:

print("There are no common fruits between both salesmen.")

elif option == '2':

# List extra fruits with both salesmen (fruits that are not common)

extra\_fruits\_salesman1 = Fruit\_Salesman1.difference(Fruit\_Salesman2)

extra\_fruits\_salesman2 = Fruit\_Salesman2.difference(Fruit\_Salesman1)

print(f"Extra fruits with Salesman 1: {', '.join(extra\_fruits\_salesman1)}")

print(f"Extra fruits with Salesman 2: {', '.join(extra\_fruits\_salesman2)}")

elif option == '3':

# List total fruits with both salesmen (all unique fruits)

total\_fruits = Fruit\_Salesman1.union(Fruit\_Salesman2)

print(f"Total unique fruits with both salesmen: {', '.join(total\_fruits)}")

elif option == '4':

print("Thank you for using. Goodbye!")

break

else:

print("Invalid option, please try again.")