

```
# A. Section 1 & 2

# Input/Output
emp_id = int(input("Enter Employee ID: "))
emp_name = input("Enter Employee Name: ")
monthly_salary = float(input("Enter Monthly Salary: "))
tot_deductions = float(input("Enter Total Deductions: "))
tot_allowances = float(input("Enter Total Allowances: "))

salary_in_hand = (monthly_salary + tot_allowances) - tot_deductions

print(f"\nEmployee Name: {emp_name}")
print(f"Salary in hand: {salary_in_hand}\n")

# if Conditions
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
num3 = int(input("Enter third number: "))

max_value = max(num1, num2, num3)
print(f"Maximum of the three numbers: {max_value}")

min_value = min(num1, num2, num3)
print(f"Minimum of the three numbers: {min_value}\n")

# loops
''' 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 '''
numbers = []
count = 0
total_sum = 0
maximum = None
minimum = None

while True:
    num = int(input("Enter an integer (or -1 to stop): "))

    if num == -1:
        break

    numbers.append(num)

    total_sum += num

    if maximum is None or num > maximum:
```

```

        maximum = num

    if minimum is None or num < minimum:
        minimum = num
    count += 1

print(f"Sum of all integers: {total_sum}")

if count > 0:
    average = total_sum / count
    print(f"Average of all integers: {average}")
else:
    print("No numbers entered.")

if count > 0:
    print(f"Maximum integer: {maximum}")

if count > 0:
    print(f"Minimum integer: {minimum}")

''' 2. Accept a String from User an do the following :
    1. Find the Length
    2. Display String in reverse
    3. Display every alternate Character in Upper Case
    4. Find out No of Vowels in the String
    5. Accept Username and Date of Birth (dd-mon-yy) from User
    Create a Password String which will be combination of
    1st 4 letters of username and last 2digits of Date of Birth
    followed by $ sign
    6. Encrypt the String and return Encrypted String '''

user_string = input("\nEnter a string: ")

print(f"Length of the string: {len(user_string)}")

print(f"Reversed string: {user_string[::-1]}")

alt_upper = ''.join([char.upper() if i % 2 == 0 else char for i, char in
enumerate(user_string)])
print(f"String with alternate characters in uppercase: {alt_upper}")

vowels = 'aeiouAEIOU'
vowel_count = sum(1 for char in user_string if char in vowels)
print(f"Number of vowels in the string: {vowel_count}")

username = input("Enter Username: ")
dob = input("Enter Date of Birth (dd-mon-yy): ")

```

```

password = username[:4] + dob[-2:] + "$"
print(f"Generated password: {password}")

def encrypt_string(s):
    encrypted = ''.join([chr(ord(char) + 3) for char in s])
    return encrypted

encrypted_password = encrypt_string(password)
print(f"Encrypted password: {encrypted_password}")

''' 3. Write Python Program to do the following :
    1. Display Area of:
        Circle
        Parallelogram '''
import math

radius = float(input("\nEnter radius of circle: "))
area_circle = math.pi * radius ** 2
print(f"Area of circle: {area_circle:.2f}")

base = float(input("Enter base of parallelogram: "))
height = float(input("Enter height of parallelogram: "))
area_parallelogram = base * height
print(f"Area of parallelogram: {area_parallelogram:.2f}")

num = int(input("\nEnter an integer to find square root: "))
sqrt_num = math.sqrt(num)
print(f"Square root of {num}: {sqrt_num:.2f}")

# B. Session 3/4
''' 1. Create a List for the following :
    a. Accept Fruits Name and their price(per kg)
    b. Fruits Name should be at odd index position in the List.
        Price at even index position '''

fruits = []

while True:
    fruit_name = input("Enter fruit name (or type 'STOP' to finish): ")
    if fruit_name.upper() == 'STOP':
        break
    fruit_price = float(input(f"Enter price for {fruit_name} (per kg): "))

    fruits.append(fruit_name)
    fruits.append(fruit_price)

```

```

''' 2. Customer will buy fruits from you (Show him the Fruits Menu)
Write a Program to
a. Calculate Total Price of Fruits Bought .
(Assume price for 1 kg )
b. Add New Fruits in the List
c. Show Total Fruits in the List '''

print("\nFruits Menu:")
for i in range(0, len(fruits), 2):
    print(f"{i//2 + 1}. {fruits[i]}: Rs. {fruits[i+1]}/kg")

total_price = 0
while True:
    fruit_choice = input("Enter the fruit name you want to buy (or type 'STOP'
to finish): ")
    if fruit_choice.upper() == 'STOP':
        break
    if fruit_choice in fruits:
        price_index = fruits.index(fruit_choice) + 1
        total_price += fruits[price_index]
    else:
        print("Fruit not available in the menu.")

print(f"\nTotal price of fruits bought: Rs. {total_price}")

new_fruit = input("\nEnter new fruit name to add: ")
new_price = float(input(f"Enter price for {new_fruit} (per kg): "))
fruits.append(new_fruit)
fruits.append(new_price)

print("\nTotal Fruits in the list:")
for i in range(0, len(fruits), 2):
    print(f"{i//2 + 1}. {fruits[i]}: Rs. {fruits[i+1]}/kg")

''' 3. Create Foll. Information in the Tuple (atleast 5 Employees)
1. EmpId - Phone Numbers (One Employee can have Multiple Numbers )
2. Accept Empid from User.
Display his Numbers only if he exists in the Database(Tuple)
Display App. Message if not present
3. Update Employee phone Number
Accept Empid from User
Check whether he / she Exists
Accept New Phone Number
Update
Display Appropriate Message for any task '''

employee_info = (
    (101, ['9876543210', '8765432109']),

```

```

(102, ['9123456789']),
(103, ['9988776655', '8899776655']),
(104, ['9000123456']),
(105, ['7890123456', '7896541230'])
)

emp_id = int(input("\nEnter Employee ID to find phone numbers: "))
found = False
for emp in employee_info:
    if emp[0] == emp_id:
        found = True
        print(f"Phone Numbers for Employee {emp_id}: {' '.join(emp[1])}")
        break
if not found:
    print("Employee not found.")

emp_id = int(input("\nEnter Employee ID to update phone number: "))
new_phone = input("Enter the new phone number: ")

found = False
for emp in employee_info:
    if emp[0] == emp_id:
        found = True
        emp[1].append(new_phone)
        print(f"Updated Phone Numbers for Employee {emp_id}: {' '.join(emp[1])}")
        break
if not found:
    print("Employee not found.")

''' 4. Store the Following info in Dictionary
    Department Name and their Employee Names
    Note : One Department can have multiple Employees

    Perform the Following Operations :
    1. Add a New Department Name and Employees in that Department
    If a New Department Name doesnot Exists
    2. Accept Dept Name from User and List all Employees
    If Dept Name Exists in the Database
    3. Add a New Employee in Existing Department
    4. Delete Existing Employee From Department '''

departments = {
    'HR': ['Alice', 'Bob'],
    'IT': ['Charlie', 'David'],
    'Sales': ['Eve', 'Frank']
}

```

```

# 1. Add new department and employees
new_dept = input("\nEnter new department name: ")
if new_dept not in departments:
    new_employees = input(f"Enter employees for {new_dept} (comma separated): ")
    new_employees = new_employees.split(',')
    departments[new_dept] = new_employees
else:
    print(f"{new_dept} already exists.")

# 2. List all employees in a department
dept_name = input("\nEnter department name to list all employees: ")
if dept_name in departments:
    print(f"Employees in {dept_name}: {' '.join(departments[dept_name])}")
else:
    print(f"{dept_name} does not exist.")

# 3. Add a new employee in an existing department
dept_name = input("\nEnter department name to add a new employee: ")
if dept_name in departments:
    new_employee = input("Enter the new employee name: ")
    departments[dept_name].append(new_employee)
    print(f"Updated employees in {dept_name}: {' '.join(departments[dept_name])}")
else:
    print(f"{dept_name} does not exist.")

# 4. Delete existing employee from department
dept_name = input("\nEnter department name to delete an employee from: ")
if dept_name in departments:
    emp_name = input("Enter the employee name to delete: ")
    if emp_name in departments[dept_name]:
        departments[dept_name].remove(emp_name)
        print(f"Updated employees in {dept_name}: {' '.join(departments[dept_name])}")
    else:
        print(f"{emp_name} not found in {dept_name}.")
else:
    print(f"{dept_name} does not exist.")

''' 5. Create Following two Sets
    1. Fruit_Salesman1
    2. Fruit_Salesman2
    Create Fruits for both Salesmans
    Perform the Following Operations
    1. Find out Common Fruits with both Salesman
    2. List Extra Fruits with Both Salesman
'''

```

3. List Total Fruits with both Salesman '''

```
salesman1 = {'apple', 'banana', 'cherry', 'dates'}
salesman2 = {'banana', 'cherry', 'figs', 'grapes'}

# 1. Find common fruits between both salesmen
common_fruits = salesman1 & salesman2
print(f"Common fruits: {'', '.join(common_fruits)}")

# 2. List extra fruits with both salesmen
extra_salesman1 = salesman1 - salesman2
extra_salesman2 = salesman2 - salesman1
print(f"Extra fruits with salesman 1: {'', '.join(extra_salesman1)}")
print(f"Extra fruits with salesman 2: {'', '.join(extra_salesman2)}")

# 3. List total fruits with both salesmen
total_fruits = salesman1 | salesman2
print(f"Total fruits sold by both salesmen: {'', '.join(total_fruits)}")
```