```
# SECTION A
# 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: "))
ename=input("enter 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("Employee name is {} and salary in hand is
{}".format(ename,salary_in_hand))


# if Conditions :
# 1. Accept 3 integers from the User and Display Maximum
num1=int(input("enter num1: "))
num2=int(input("enter num2: "))
num3=int(input("enter num3: "))
if num1>num2 and num1>num3:
    print("num1",num1,"is maximum")
elif num2>num3: #num2 > num1 condition is no need to check because it checked in
above if condition
    print("num2",num2,"is maximum")
else:
    print("num3",num3,"is maximum")
# 2. Accept 3 integers from USer and display Minimum
num1=int(input("enter num1: "))
num2=int(input("enter num2: "))
num3=int(input("enter num3: "))
if num1<num2 and num1<num3:
    print("num1",num1,"is minimum")
elif num2<num3: #num2 > num1 condition is no need to check because it checked in
above if condition
    print("num2",num2,"is minimum")
else:
    print("num3",num3,"is minimum")



# 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
sum=0
count=0
max=0
min=0
while True:
    user_input=input("Enter integer: ")

    if user_input=='exit' or user_input=='end' or user_input=='quit':
        break
    user_input=int(user_input)
    sum+=user_input
    count+=1
    if max<user_input:
        max=user_input
    if min>user_input:
        min=user_input
```

```
print("The sum of integers is:",sum)
print("Average of integers:",sum/count)
print("Maximum integer from all is ",max)
print("Minimum integer from all ",min)



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

string=input("enter string: ")
# 1. Find the Length
length=len(string)
print(length)
# 2. Display String in reverse
print("String in reverse:",string[::-1])
# 3. Display every alternate Character in Upper Case
print("Alternate Character from String in UpperCase: ",string[::2].upper())
# 4. Find out No of Vowels in the String
vowels='aeiouAEIOU'
count=0
for i in string:
    if i in vowels:
        count+=1
print("No of vowels in string :",count)
# 5. Accept Username and Date of Birth (dd-mon-yy) from User Create a Password
String which will be combination of
username=input("enter username:")
Date_of_Birth=input("Enter data of birth(dd-mon-yy):")
if len(username)>=4 and len(Date_of_Birth)==8:
    password=username[:4]+Date_of_Birth[-2:]
    print("Generate password:",password)
    # 6. Encrypt the String and return Encrypted String
    shift_value=3
    encrypted_password=''
    for i in password:
        if i.isupper():
            encrypted_password+=chr((ord(i) + shift_value - 65) % 26 + 65)
        elif i.islower():
            encrypted_password+=chr((ord(i) + shift_value - 97) % 26 + 97)
        else:
            encrypted_password += i
    print("encrypted password: ",encrypted_password)
else:
    print("Invalid Input")




# 3. Write Python Program to do the following :
     # 1. Display Area of
      # Circle
      # Parallelogram
pi=3.14
radius=float(input("Enter float value: "))
circle_area=pi*radius**2
print("Area of Circle: ",circle_area)
base = float(input("Enter the base of the parallelogram: "))
height = float(input("Enter the height of the parallelogram: "))
parallelogram_area = base * height
print("Area of parallelogram: ",parallelogram_area)

# 4. Accept Integer and find Square root of Integer
integer=int(input())
print("Square root of",integer,"is",integer**0.5)
```

```
# SECTION B

# 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 names:")
    if fruit_name=='exit' or fruit_name=='end' or fruit_name=='quit':
        break
    fruit_price=float(input("Enter fruit price: "))
    fruits.append(fruit_price)
    fruits.append(fruit_name)

print("Fruits Availablle are :",fruits)


# 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
while True:
    print("\n1. Fruits Menu")
    print("2. Calculate Total Price of Fruits Bought")
    print("3. Add New Fruits in the List")
    print("4. Show Total Fruits in the List")
    print("5. Exit")
    choice = int(input("Enter your choice: "))
    if choice==1:
        print("Fruits Menu: ")
        for i in range(0,len(fruits),2):
            print(fruits[i+1],"--->",fruits[i])
    elif choice==2:
        total=0
        for i in range(0,len(fruits),2):
            total=total+fruits[i]
        print("total price of fruits: ",total)
    elif choice == 3:
        # Add New Fruits in the List
        new_fruit = input("Enter new fruit name: ")
        new_price = float(input(f"Enter price per kg for {new_fruit}: "))
        fruits.append(new_price)
        fruits.append(new_fruit)

    elif choice == 4:
        # Show Total Fruits in the List
        print("Total fruits in the list:")
        for i in range(1, len(fruits), 2):
            print(fruits[i])
    elif choice == 5:
        break
    else:
        print("invalid")

# 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_data = (
    (101, ['9876543210', '8765432109']),
    (102, ['7654321098']),
    (103, ['6543210987', '5432109876']),
    (104, ['4321098765']),
    (105, ['3210987654', '2109876543'])
)
emp_id = int(input("Enter Employee ID to display phone numbers: "))
for emp in employee_data:
    if emp[0] == emp_id:
        print("Phone number of employee whose id is {} and their numbers are 
{}".format(emp_id,emp[1]))
        break
else:
    print("Employee ID not found in the database.")
emp_id = int(input("Enter Employee ID to update phone number: "))
for emp in employee_data:
    if emp[0]==emp_id:
        new_number=input("Enter new number: ")
        emp[1].append(new_number)
        print("Updated phone numbers for employee", emp_id, ":", emp[1])
        break
else:
    print("Employee ID not found in the database.")




# 4. Store the Following info in Dictionary

department_data = {
    "HR": ["preethi", "pooja"],
    "IT": ["Chaitanya", "ramu"],
    "Finance": ["sreeja", "pooja"]
}
# 1. Add a New Department Name and Employees in that Department
dept_name = "Marketing"
employees = ["Gowri", "Hanuman"]
if dept_name not in department_data:
    department_data[dept_name] = employees
print("Added new department { } with employees 
{}".format(dept_name,department_data[dept_name]))

# 2. Accept Dept Name from User and List all Employees
# If Dept Name Exists in the Database
dept_name = input("enter dept name:")
if dept_name in department_data:
    print(f"Employees in {dept_name}: {department_data[dept_name]}")
else:
    print(f"Department {dept_name} not found.")
# 3. Add a New Employee to an Existing Department
dept_name = "Finance"
if dept_name in department_data:
    new_employee = input("enter new employee:")
    department_data[dept_name].append(new_employee)
    print(f"Added {new_employee} to {dept_name}. Updated list: 
```

```
{department_data[dept_name]}")
else:
    print(f"Department {dept_name} not found.")
# 4. Delete Existing Employee From Department
dept_name = "HR"
if dept_name in department_data:
    employee_to_delete = "preethi"
    if employee_to_delete in department_data[dept_name]:
        department_data[dept_name].remove(employee_to_delete)
        print(f"Removed {employee_to_delete} from {dept_name}. Updated list:
{department_data[dept_name]}")
    else:
        print(f"Employee {employee_to_delete} not found in {dept_name}.")
else:
    print(f"Department {dept_name} not found.")



# 5. Create Following two Sets
# 1. Fruit_Salesman1
# 2. Fruit_Salesman2
# Create Fruits for both Salesmans

Fruit_Salesman1 = {"Apple", "Banana", "Mango", "Orange", "Pineapple"}
Fruit_Salesman2 = {"Banana", "Grapes", "Orange", "Watermelon", "Mango"}

# 1. Find out Common Fruits with both Salesmen (Intersection using & operator)
common_fruits = Fruit_Salesman1 & Fruit_Salesman2
print("Common Fruits sold by both Salesmen:", common_fruits)

# 2. List Extra Fruits with Both Salesmen (Symmetric Difference using ^
operator)
extra_fruits = Fruit_Salesman1 ^ Fruit_Salesman2
print("Extra Fruits sold by either Salesman:", extra_fruits)

# 3. List Total Fruits with both Salesmen (Union using | operator)
total_fruits = Fruit_Salesman1 | Fruit_Salesman2
print("Total Fruits sold by both Salesmen:", total_fruits)
```