TASK-1:TO DO LIST

```
class ToDoList:
    def __init__(self):
        self.tasks = []
    def add_task(self,task):
        self.tasks.append({"task": task, "completed": False})
    def view_tasks(self):
        if not self.tasks:
            print('Your To-Do list is empty')
        else:
            print('Your To-Do List:')
            for index, task in enumerate(self.tasks, start=1):
                status = "✓" if task["completed"] else "○"
                print(f"{index}. [{status}] {task['task']}")
    def mark_complete(self,task_index):
        if 1 <= task_index <= len(self.tasks):
            self.tasks[task_index - 1]['completed'] = True
            print(f'Task {task_index} marked as complete')
        else:
            print('Invalid task number')
    def remove_task(self,task_index):
        if 1 <= task_index <= len(self.tasks):
            del self.tasks[task_index - 1]
            print(f'Task {task_index} removed from the list.')
        else:
            print('Invalid task number')
    def menu(self):
        while True:
            print('List of operations:')
            print('1.Add a task')
            print('2.View all tasks')
            print('3.Mark task as Complete')
            print('4.Remove a task')
            print('5.Exit')
            choice = input('Enter your choice from 1 to 5: ')
            if choice == '1':
                task = input('Enter task: ')
                self.add_task(task)
            elif choice == '2':
                self.view_tasks()
            elif choice == '3':
                task_index = int(input('Enter task number to mark as complete: '))
                self.mark_complete(task_index)
            elif choice == '4':
                task_index = int(input("Enter task number to remove: "))
                self.remove_task(task_index)
            elif choice == '5':
                print('Exiting.')
                break
            else:
                print('Invalid choice.Please enter a number.')
if __name__ == "__main__":
    to_do_list = ToDoList()
    to_do_list.menu()
```

TASK-2:SIMPLE CALCULATOR

```
def add(a,b):
    return a+b
def subtract(a,b):
    return a-b
def multiply(a,b):
    return a*b
def divide(a,b):
    if b == 0:
        print('Zero division Error')
    else:
        return a/b
print('Simple Calculator')
print('Available operations are:')
print('1.Addition')
print('2.Subtraction')
print('3.Multiplication')
print('4.Division')
x=float(input('Enter the first number:'))
y=float(input('Enter the second number:'))
operation=input('Choose an operation from 1/2/3/4:')
if operation == '1':
    res = add(x,y)
    print(f'{x}+{y} = {res}')
```

```
elif operation == '2':
    res = subtract(x,y)
    print(f'{x}-{y} = {res}')
elif operation == '3':
    res = multiply(x,y)
    print(f'{x}*{y} = {res}')
elif operation == '4':
    res = divide(x, y)
    print(f'{x}/{y} = {res}')
else:
    print('Invalid operation.Please choose a valid operation.')
```

TASK-3:PASSWORD GENERATOR

```
import random
import string
def generate_password(length):
    lowercase_letters = string.ascii_lowercase
    uppercase_letters = string.ascii_uppercase
    digits = string.digits
    symbols = string.punctuation
    all_characters = lowercase_letters + uppercase_letters + digits + symbols
    password = ''.join(random.choices(all_characters,k=length))
    return password
if __name__ == "__main__":
    try:
        password_length = int(input("Enter the desired length of the password: "))
        generated_password = generate_password(password_length)
        print(f'Generated Password: {generated_password}')
    except ValueError:
        print('Invalid input.Please enter a valid integer')
```

TASK-4:CONTACT BOOK

```
class ContactBook:
    def _init_(self):
        self.contacts = []
    def add_contact(self, name, phone, email, address):
        contact = {
            'name': name,
            'phone': phone,
            'email': email,
            'address': address
        }
        self.contacts.append(contact)
        print(f"Contact '{name}' added successfully.")
    def view_contacts(self):
        if not self.contacts:
            print("Contact list is empty.")
        else:
            print("Contact List:")
            for idx, contact in enumerate(self.contacts, start=1):
                print(f"{idx}. Name: {contact['name']}, Phone: {contact['phone']}")
    def search_contact(self, query):
        found_contacts = []
        for contact in self.contacts:
            if query.lower() in contact['name'].lower() or query in contact['phone']:
                found_contacts.append(contact)
        if found_contacts:
            print(f"Found {len(found_contacts)} contact(s) matching '{query}':")
            for contact in found_contacts:
                print(f"Name: {contact['name']}, Phone: {contact['phone']}, Email: {contact['email']}, Address: {contact['address']}")
        else:
            print(f"No contacts found matching '{query}'.")
    def update_contact(self, name):
        for contact in self.contacts:
            if contact['name'].lower() == name.lower():
                print(f"Current details for '{name}':")
                print(f"Phone: {contact['phone']}, Email: {contact['email']}, Address: {contact['address']}")
                contact['phone'] = input("Enter new phone number: ")
                contact['email'] = input("Enter new email address: ")
                contact['address'] = input("Enter new address: ")
                print(f"Contact '{name}' updated successfully.")
                return
        print(f"Contact '{name}' not found.")
    def delete_contact(self, name):
        for idx, contact in enumerate(self.contacts):
            if contact['name'].lower() == name.lower():
                del self.contacts[idx]
                print(f"Contact '{name}' deleted successfully.")
                return
        print(f"Contact '{name}' not found.")
def main():
    contact_book = ContactBook()
    while True:
        print("\nContact Book Menu:")
        print("1. Add Contact")
        print("2. View Contacts")
        print("3. Search Contact")
        print("4. Update Contact")
        print("5. Delete Contact")
        print("6. Quit")
        choice = input("Enter your choice (1-6): ")
        if choice == '1':
            name = input("Enter name: ")
            phone = input("Enter phone number: ")
            email = input("Enter email address: ")
            address = input("Enter address: ")
            contact_book.add_contact(name, phone, email, address)
        elif choice == '2':
            contact_book.view_contacts()
        elif choice == '3':
            query = input("Enter name or phone number to search: ")
            contact_book.search_contact(query)
        elif choice == '4':
            name = input("Enter name of contact to update: ")
            contact_book.update_contact(name)
        elif choice == '5':
            name = input("Enter name of contact to delete: ")
            contact_book.delete_contact(name)
        elif choice == '6':
            print("Exiting Contact Book. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number from 1 to 6.")

if __name__ == "__main__":
    main()
```