CODSOFT INTERNSHIP

DOMAIN: PYTHON PROGRAMMING

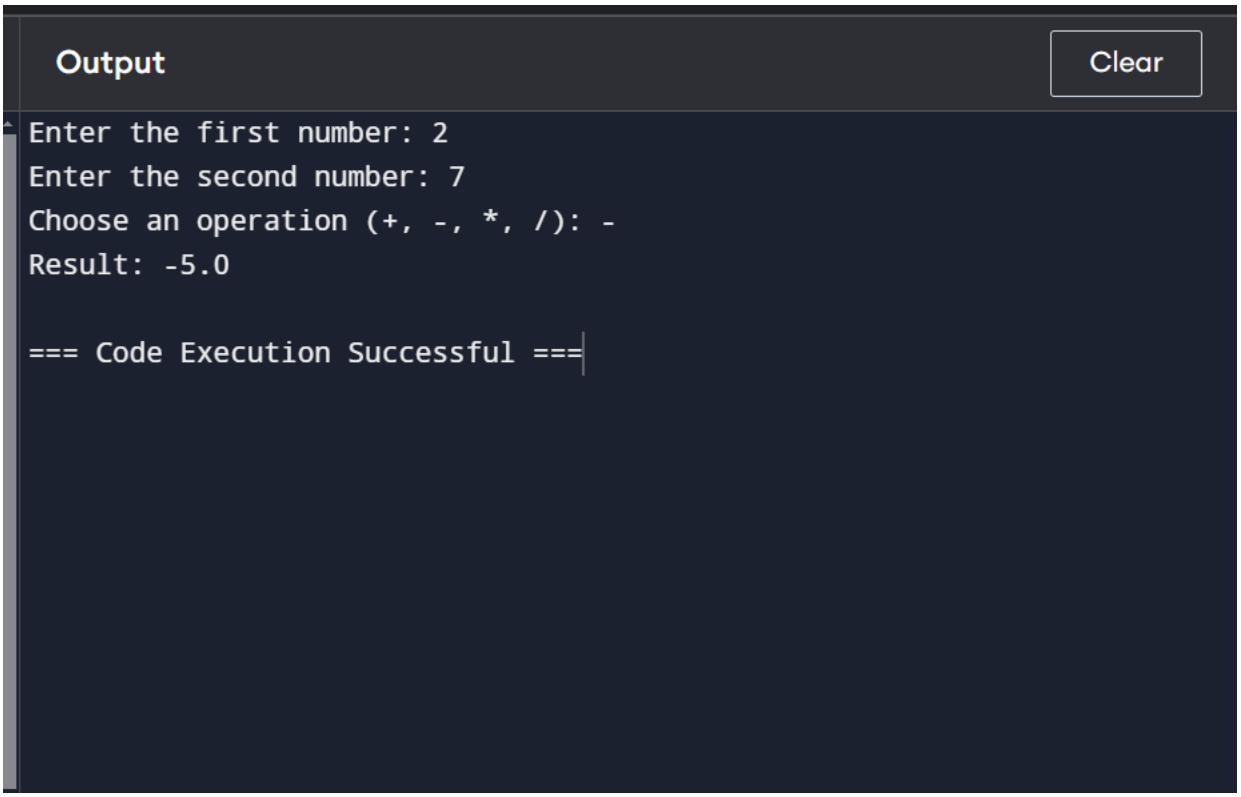
TASK –1

SIMPLE CALCULATOR

```
# Function to perform basic arithmetic operations
def calculator(num1, num2, operation):
    if operation == '+':
        return num1 + num2
    elif operation == '-':
        return num1 - num2
    elif operation == '*':
        return num1 * num2
    elif operation == '/':
        # Check for division by zero
        if num2 != 0:
            return num1 / num2
        else:
            return "Error: Division by zero!"
    else:
        return "Invalid operation!"

# Prompt user to enter two numbers
try:
    num1 = float(input("Enter the first number: "))
```

```
    num2 = float(input("Enter the second number: "))

    # Prompt user to choose an operation
    operation = input("Choose an operation (+, -, *, /): ")

    # Perform calculation and display the result
    result = calculator(num1, num2, operation)
    print("Result:", result)

except ValueError:
    print("Error: Invalid input! Please enter numeric values.")
```

OUTPUT:



TASK -2

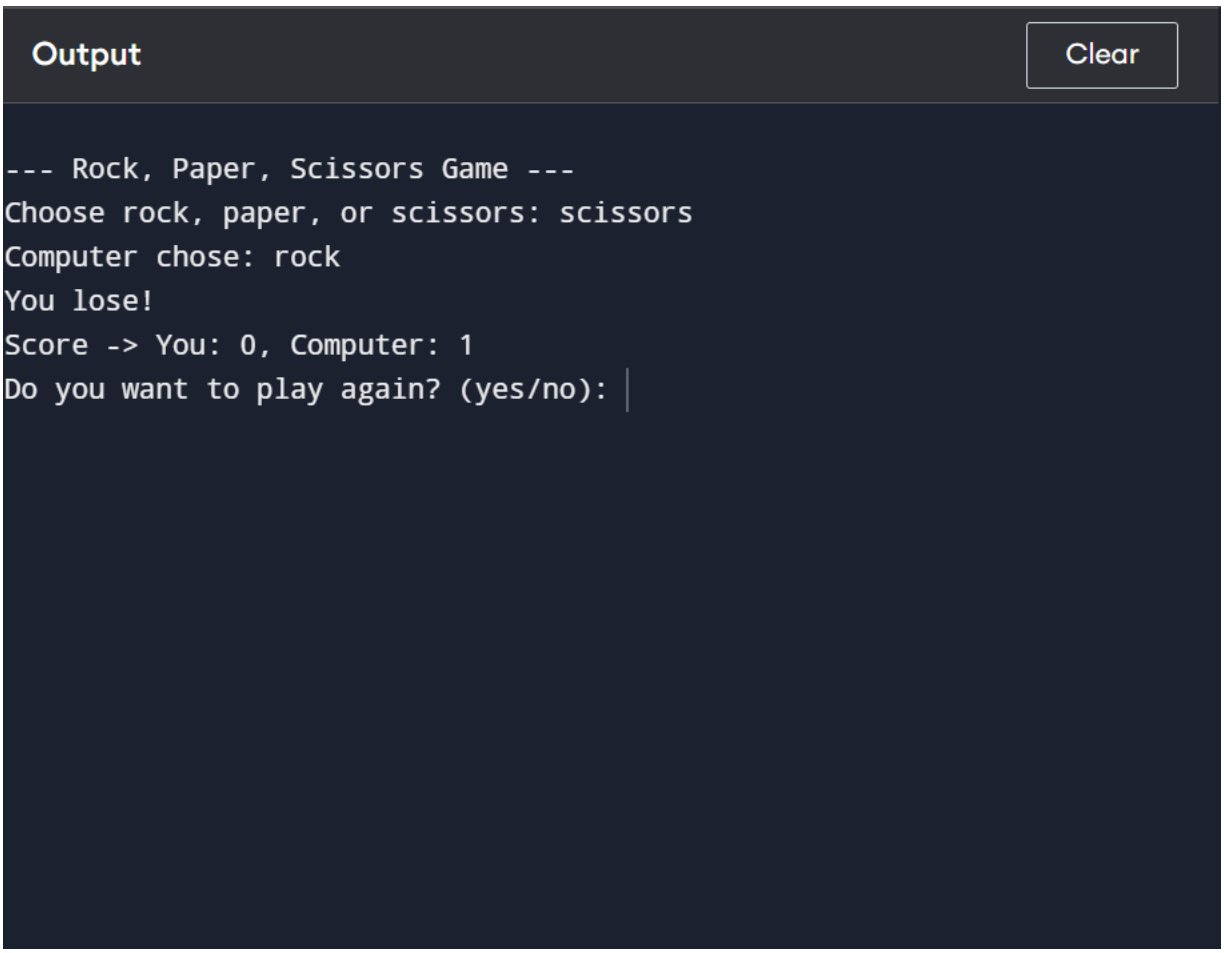
ROCK, PAPER,SCISSOR, GAME

```
import random


def get_computer_choice():
    return random.choice(["rock", "paper", "scissors"])


def determine_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return "It's a tie!"
```

```
    elif (user_choice == "rock" and computer_choice == "scissors") or \
         (user_choice == "scissors" and computer_choice == "paper") or \
         (user_choice == "paper" and computer_choice == "rock"):
        return "You win!"
    else:
        return "You lose!"

def play_game():
    user_score = 0
    computer_score = 0

    while True:
        print("\n--- Rock, Paper, Scissors Game ---")
        user_choice = input("Choose rock, paper, or scissors: ").lower()
        if user_choice not in ["rock", "paper", "scissors"]:
            print("Invalid choice. Please choose rock, paper, or scissors.")
            continue

        computer_choice = get_computer_choice()
        print(f"Computer chose: {computer_choice}")

        result = determine_winner(user_choice, computer_choice)
        print(result)

        if result == "You win!":
            user_score += 1
```

```
    elif result == "You lose!":
        computer_score += 1

    print(f"Score -> You: {user_score}, Computer: {computer_score}")

    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again != "yes":
        print("Thanks for playing!")
        Break
     play_game()
```

OUTPUT:



TASK-3

CONTACT BOOK:

class Contact:

```
    def __init__(self, name, phone, email, address):
        self.name = name
        self.phone = phone
        self.email = email
        self.address = address

class ContactManager:
    def __init__(self):
        self.contacts = []

    def add_contact(self, name, phone, email, address):
        contact = Contact(name, phone, email, address)
        self.contacts.append(contact)
        print(f"Contact '{name}' added successfully.")

    def view_contacts(self):
        if not self.contacts:
            print("No contacts found.")
            return
        print("Contact List:")
        for i, contact in enumerate(self.contacts, 1):
            print(f"{i}. Name: {contact.name}, Phone: {contact.phone}")

    def search_contact(self, search_term):
        results = [contact for contact in self.contacts if search_term in contact.name or
search_term in contact.phone]
```

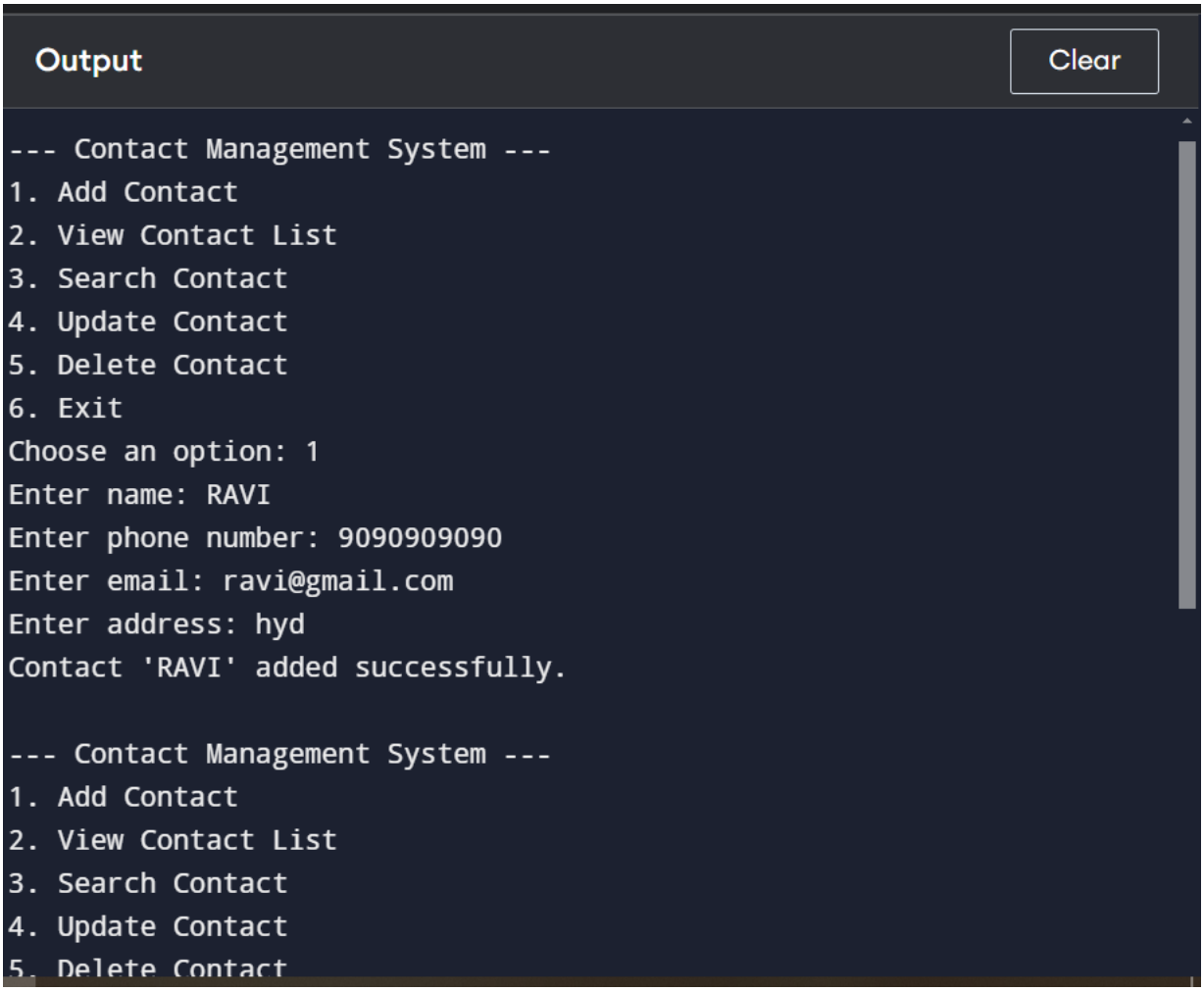
```
        if results:
            for contact in results:
                self.display_contact(contact)
        else:
            print("No contacts found.")

    def update_contact(self, search_term, new_name=None, new_phone=None, new_email=None, new_address=None):
        for contact in self.contacts:
            if search_term in contact.name or search_term in contact.phone:
                if new_name:
                    contact.name = new_name
                if new_phone:
                    contact.phone = new_phone
                if new_email:
                    contact.email = new_email
                if new_address:
                    contact.address = new_address
                print(f"Contact '{contact.name}' updated successfully.")
                return
        print("Contact not found.")

    def delete_contact(self, search_term):
        for contact in self.contacts:
            if search_term in contact.name or search_term in contact.phone:
                self.contacts.remove(contact)
```

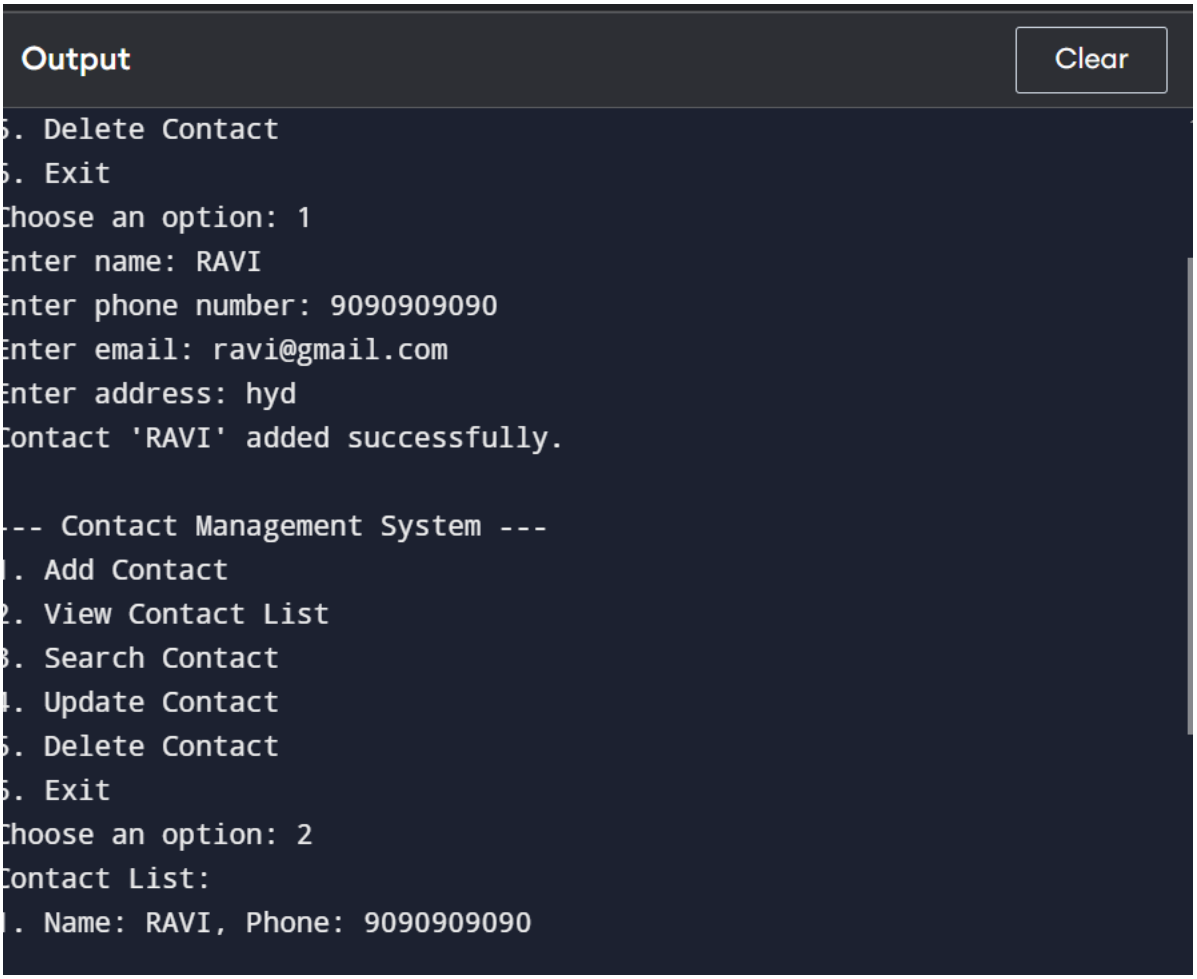
```
            print(f"Contact '{contact.name}' deleted successfully.")
            return
        print("Contact not found.")

    def display_contact(self, contact):
        print(f"Name: {contact.name}")
        print(f"Phone: {contact.phone}")
        print(f"Email: {contact.email}")
        print(f"Address: {contact.address}")

def menu():
    manager = ContactManager()

    while True:
        print("\n--- Contact Management System ---")
        print("1. Add Contact")
        print("2. View Contact List")
        print("3. Search Contact")
        print("4. Update Contact")
        print("5. Delete Contact")
        print("6. Exit")
        choice = input("Choose an option: ")

        if choice == '1':
            name = input("Enter name: ")
            phone = input("Enter phone number: ")
```

```
            email = input("Enter email: ")
            address = input("Enter address: ")
            manager.add_contact(name, phone, email, address)
        elif choice == '2':
            manager.view_contacts()
        elif choice == '3':
            search_term = input("Enter name or phone number to search: ")
            manager.search_contact(search_term)
        elif choice == '4':
            search_term = input("Enter name or phone number to update: ")
            new_name = input("Enter new name (leave blank to keep current): ") or None
            new_phone = input("Enter new phone number (leave blank to keep current): ") or
None
            new_email = input("Enter new email (leave blank to keep current): ") or None
            new_address = input("Enter new address (leave blank to keep current): ") or None
            manager.update_contact(search_term, new_name, new_phone, new_email,
new_address)
        elif choice == '5':
            search_term = input("Enter name or phone number to delete: ")
            manager.delete_contact(search_term)
        elif choice == '6':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
```

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

OUTPUT:




Output
Clear
5. Delete Contact
6. Exit
Choose an option: 1
Enter name: RAVI
Enter phone number: 9090909090
Enter email: ravi@gmail.com
Enter address: hyd
Contact 'RAVI' added successfully.
--- Contact Management System ---
1. Add Contact
2. View Contact List
3. Search Contact
4. Update Contact
5. Delete Contact
6. Exit
Choose an option: 2
Contact List:
1. Name: RAVI, Phone: 9090909090