class Volunteer:

def \_\_init\_\_(self, name, age, skills):

self.name = name

self.age = age

self.skills = skills

class Task:

def \_\_init\_\_(self, name, required\_skills):

self.name = name

self.required\_skills = required\_skills

self.assigned\_volunteer = None

class VolunteerManagementSystem:

def \_\_init\_\_(self):

self.volunteers = []

self.tasks = []

def add\_volunteer(self, name, age, skills):

volunteer = Volunteer(name, age, skills)

self.volunteers.append(volunteer)

def add\_task(self, name, required\_skills):

task = Task(name, required\_skills)

self.tasks.append(task)

def assign\_task(self, task\_name):

for task in self.tasks:

if task.name == task\_name and task.assigned\_volunteer is None:

for volunteer in self.volunteers:

if set(task.required\_skills).issubset(set(volunteer.skills)):

task.assigned\_volunteer = volunteer

return

print("No suitable volunteer found for task")

system = VolunteerManagementSystem()

name=input("Enter name: ")

age=input("Enter age: ")

skill=input("Enter skills: ")

system.add\_volunteer(name,age,[skill])

system.add\_task("Coding Challenge", ["Problem-Solving"])

system.assign\_task("Coding Challenge")

print("Task is Assigned to: ",system.tasks[0].assigned\_volunteer)