







# Expert system - Employee information management

# functions:

# add employee

# remove employee

# work start time

# work end time

# update work progress

# get work progress

# employee class

class Employee:

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

        self.inTime = "-1"

        self.outTime = "-1"

        self.workProgress = 0

    def printDetails(self):

        print("Id: ", self.id)

        print("Name: ", self.name)

        if (self.inTime == "-1"):

            print(self.name+" not started work yet.")

        else:

            print(self.name+" not started work at " + self.inTime)

        if (self.outTime == "-1"):

            print(self.name+" not completed work yet.")

        else:

            print(self.name+" completed work at " + self.outTime)

        print("Work Progress: ", self.workProgress)

    def updateWorkProgress(self, addition):

        self.workProgress += addition

        self.workProgress = min(self.workProgress, 100)

    def updateInTime(self, inTime):

        self.inTime = inTime

    def updateOutTime(self, outTime):

        self.outTime = outTime

class EmployeeManagement:

    def \_\_init\_\_(self):

        self.employees = []

        self.employeesCnt = 0

    def addEmployee(self, name):

        self.employeesCnt += 1

        employee = Employee(self.employeesCnt, name)

        self.employees.append(employee)

        return self.employeesCnt

    def getEmployeeInd(self, id):

        for i in range(len(self.employees)):

            if (self.employees[i].id == id):

                return i

        return -1

    def removeEmployee(self, id):

        ind = self.getEmployeeInd(id)

        if (ind == -1):

            print("The employee with " + id + " does not exist")

            return

        self.employees.pop(ind)

    def addWorkProgress(self, id, amount):

        ind = self.getEmployeeInd(id)

        if (ind == -1):

            print("The employee with " + id + " does not exist")

            return

        self.employees[ind].updateWorkProgress(amount)

    def startWork(self, id, startTime):

        ind = self.getEmployeeInd(id)

        if (ind == -1):

            print("The employee with " + id + " does not exist")

            return

        self.employees[ind].updateInTime(startTime)

    def endWork(self, id, endTime):

        ind = self.getEmployeeInd(id)

        if (ind == -1):

            print("The employee with " + id + " does not exist")

            return

        self.employees[ind].updateInTime(endTime)

    def printAllEmployees(self):

        for employee in self.employees:

            employee.printDetails()

            print()

def main():

    system = EmployeeManagement()

    while (1):

        print("\n\nWhat you want to do?")

        print("1 Add Employee")

        print("2 Remove Employee")

        print("3 Add work start time")

        print("4 Add work end time")

        print("5 Add work progress")

        print("6 Print all employees in the system")

        print("0 Exit")

        choice = int(input(": "))

        if (choice == 1):

            name = input("Enter Name: ")

            id = system.addEmployee(name)

            print("Employee added with id: ", id)

        elif (choice == 2):

            name = input("Enter Name: ")

            system.removeEmployee(name)

        elif (choice == 3):

            id = int(input("Enter id: "))

            startTime = input("Enter Start Time: ")

            system.startWork(id, startTime)

        elif (choice == 4):

            id = int(input("Enter id: "))

            endTime = input("Enter End Time: ")

            system.startWork(id, endTime)

        elif (choice == 5):

            id = int(input("Enter id: "))

            progress = int(input("Add progress: "))

            system.addWorkProgress(id, progress)

        elif (choice == 6):

            system.printAllEmployees()

        else:

            print("Thank you!")

            break

main()