CS 4097D Object-Oriented Systems Lab: Assignment 3

1) Write a Java program which reverses a linked list. It takes input N and creates a link list having N nodes (as per the given class definition) having character data. Create a class *ReverseMain*, and define a method *reverseList* which takes the starting node as the input parameter and returns the starting node in the reversed list.
Note: Don't use array to hold the node objects

```
Class Node
{
        char data;
        Node next;
}
```

Sample Input/output

```
Enter the number of nodes: 5
Enter the data in Node1 : a
Enter the data in Node1 : b
Enter the data in Node1 : c
Enter the data in Node1 : d
Enter the data in Node1 : e
The reversed list is:
e->d->c->b->a
```

2) Implement an invoice item management system which stores a list of invoice items and calculates the total amount. Each invoice item stores the inventory item details and the quantity of that particular inventory item purchased.

Each inventory item has an id, name and unit price which should be stored prior. Implement a menu driven program which iteratively reads the invoice details, and up on completion it generates an invoice which lists the invoice details along with the total amount. While entering invoice details, only the inventory id and quantity purchased is needed.

Sample input and output:

```
Enter the available inventory items line by line, type OKAY at the end of the list:
100 Biscuit 20
102 Chocolate 30
105 Milk 10
```

```
106 Icecream 10
109 RiceInKg 45
OKAY

1.Add Invoice
2.Exit
Choice:
1

Add Invoice Item(Y/N)
Y
Enter Inventory ID and Quantity line by line, type OKAY at the end of the list
100 1
109 2
OKAY
****INVOICE****
100,Biscuit,20,1
109,RiceInKg,45,2
Total:110
********

1.Add Invoice
2.Exit
Choice:
2
```

3) Create a class DepartmentMeetingScheduler, which schedules the meetings happening in the department, and lists the meeting details scheduled on a particular day. Each meeting has the characteristics:- meetingTitle, meeting date(in dd/MM/yyyy format), meetingStartTime(in 24hr format), meetingEndTime(in 24hr format), meetingDescription.
Note: Use an ArrayList to store all meeting objects

DepartmentMeetingScheduler have the following specification:

- There is a method *scheduleMeeting*, which looks if any meeting which is already scheduled is colliding with the new meeting (based on timing) and creates the meeting if no collision occurs.
- There is a method displayMeetings which takes an input parameter date, and lists all the meetings scheduled on the given date

Sample Input/Output

```
1. Schedule Meeting
2. List Meetings

Enter your choice : 1
Title - DC Meeting
Date - 23/08/2021
Start time - 15:30
End time - 16:30
Description - first DC meeting of Manjusha
Scheduled successfully

Enter your choice : 1
Title - Class Committee Meeting
Date - 23/08/2021
Start time - 16:00
End time - 16:30
Description - first class committee meeting of first year students
Collision! Already another meeting is scheduled

Enter your choice : 2
On 23/08/2021, from 15.30 to 16.30 : DC Meeting : first DC meeting of Manjusha
```

4) Create a class Employee(id, name, department, salary) and define a subclass Manager of Employee with additional data member bonus. Now create some(n) objects of manager class and display the details of manager's having a maximum salary(including bonus). Input is given as the number of managers followed by their id, name, department, salary and bonus.

```
Input: 4
        1 Jadeja  HR  30000 3000
       2 Rishab design 40000 4500
       3 Virat  SW  42000 2500
       4 Rohit HW 38000 3500

Output:  2 Rishab design 40000 4500
          3 Virat SW 42000 2500
```

5) At the government hospital treating patients for Covid, the management is preparing a token system that gives priority to senior citizens who want to get tested over other patients. Let Patient is a subclass of Person. A person has a person_name and a patient has a patient_ID. A patient_ID have the following format: <xxxx-yyy> where xxxx is the entrycode and yyy is the patient's age. Write a java program to read details of N number of patients, and show their names in the order of age. That is, display the Name of the person who needs to be tested first and then second person and so on.

Sample input and output:

```
Enter the number of patients: 3

Enter the details of patient 1:
Name: Daniel
Patient ID: 1SDX-044

Enter the details of patient 2:
Name: Garima
Patient ID: 2SMG-029

Enter the details of student 3:
Name: Jaya
Patient ID: 1SNF-071

List of Patients to be tested are:
1SNF-071 Jaya
1SDX-044 Daniel
2SMG-029 Garima
```

6) Define a class StudentData that holds the following information of a student:
id - a String that stores that unique ID of each student
name - a String indicating the name of the student.
Note that these fields should be declared private.

Create another class StudentDataExtended with a private attribute named location (a String that stores the location of the student). In this class, define a method addDetails that stores the details of students; and a method printDetails that outputs the details of students in the sorted order of their id. (Both the classes should be in different packages).

Sample Input/output

```
Enter the number of students : 2
Enter the details of Student 1 (id, name, location):
B200 Ajay Hyderabad
Enter the details of Student 2 (id, name, location):
B100 Ramesh Hyderabad

The Student Details are:
B100 Ramesh Hyderabad
B200 Ajay Hyderabad
```