Lab 8 A

Problem 1: New York Trip

You have planned a trip to Downtown Manhattan, and marked down all the landmarks you want to visit. However on the last day before the trip, the Maps software is malfunction and you arent able to view the itinirary. Luckily you backed up the details of your route within the software. You retrieve a "input.bin" binary file that contains the location of each landmark.

As you are aware, the city can be viewed as a grid of evenly spaced perpendicular roads, each a unit distance apart. Each landmark is on the intersection of a horizontal and vertical road, and all your travel must occur via these roads. The input file contains N points, each having a x and y coordinate (stored as 4 byte integers). Your route starts on the first landmark, successively visits all landmarks and then returns to the first landmark. Return the total distance travelled modulo 1e9+7.

Note: The numbers are stored as 4 byte integers in the file. The number of landmarks should be deduced from the file contents

File Format:

- File contains N points. Each point is stored
- 4 bytes for the x coordinate
- 4 bytes for the y coordinate

Output Format:

Print 1 line of the total distance travelled (modulo 1e9+7).

Constraints:

- $1 \leq N \leq 1e6$
- $-1e9 \leq x,y \leq 1e9$

Example File:

```
1 2 3 5 8 13 (in a binary file of 24 bytes)
```

```
36
```

Problem 2: Student Grades Report

In a college administration system, the grades of students across multiple subjects are stored in a text file. The college wants a simple tool to process this data, calculate the average grade for each student, and print the list of students sorted by name.

The text file, named `students.txt`, contains records of students with the following details:

- **Student ID:** An integer representing the student's unique identification number.
- **Name:** The full name of the student (which may contain spaces).
- **Grades:** A list of integers representing the student's grades across different subjects.

Each line in the file corresponds to one student and follows the format:

```
ID, Name, Grade1, Grade2, ..., GradeN
```

Your task is to:

1. Read the student data from the `students.txt` file.
2. Calculate the average grade for each student.
3. Sort the list of students alphabetically by name.
4. Print each student's name and their average grade, rounded to two decimal places.

Constraints:

- Each line in `students.txt` represents one student and contains a valid student ID, name, and at least one grade.
- The file can contain up to 100 students.
- A student may have up to 10 grades.
- Each grade is an integer between 0 and 100.

Input Format:

The file `students.txt` contains lines formatted as follows:

```
ID, Name, Grade1, Grade2, ..., GradeN
```

Output Format:

Print each student's name and their average grade in sorted order by name. Format the output as follows:

```
Name: [Student Name], Score: [Average Grade]
```

Example File:

Suppose `students.txt` contains the following:

```
101, John Doe, 85, 90, 78, 0, 0
102, Jane Smith, 92, 88, 95, 2
103, Emily Johnson, 75, 80, 70
104, Michael Brown, 88, 94, 90
```

Example Output:

```
Name: Emily Johnson, Score: 75.00
Name: Jane Smith, Score: 69.25
Name: John Doe, Score: 50.60
Name: Michael Brown, Score: 90.67
```

Notes:

- The program reads and parses each line of data, calculates the average grade for each student, and sorts the students by name. - Only valid entries in the file are processed, and each student's average score is calculated as a floating-point number rounded to two decimal places.

Submission Guidelines

Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.