```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class School {
private String name;
private String[] address;
private String
principalFirstName;
private String
principalLastName;
private List<MarksEntry> marksEntries;

public School(String name, String[] address, String principal) {
this.name = name;
this.address = address;
this.principalFirstName = principalFirstName;
this.principalLastName = principalLastName;
this.marksEntries = new ArrayList<>();
    }

    public String getName() {
return name;
    }

    public String[] getAddress() {
return address;
    }

    public String getPrincipal() {
return principal;
    }

    public List<MarksEntry> getMarksEntries() {
        return marksEntries;
    }
```

```
public void addMarks(int year, String studentID, String studentName,
String className, double score) {

if (!marks.containsKey(year)) {
        marks.put(year, new HashMap<>());
        }
        marks.get(year).put(studentID, score);
    }

public void addMarksEntry(MarksEntry marksEntry) {
marksEntries.add(marksEntry);
    }

public double getScoreForYear(int year) {
        double totalScore = 0.0;
        int count = 0;
        for (MarksEntry entry : marksEntries)
{             if (entry.getYear() == year) {
        totalScore += entry.getScore();
count++;
           }        }
if (count == 0) {
return 0.0;         }
else {
           return totalScore / count;
        }
    }

public double getAverageScore() {
        double totalScore = 0.0;
        int count = 0;

        for (MarksEntry entry : marksEntries){
totalScore += entry.getScore();
count++;
```

```
        }
 if (count == 0) {
return 0.0;
}
else
{
            return totalScore / count;
        }
}

public double getStandardDeviation() {
        double mean = getAverageScore();
        double sumOfSquaredDeviations = 0.0;

       for (MarksEntry entry : marksEntries) {
            sumOfSquaredDeviations += Math.pow(entry.getScore() -
mean, 2);
        }
        if (marksEntries.size() == 0) {
            return 0.0;
} else {
            return Math.sqrt(sumOfSquaredDeviations /
marksEntries.size());
        }
    }

public static void sortSchools(List<School> schools) {
        Collections.sort(schools, (s1, s2) ->
        Double.compare(s2.getAverageScore(),
s1.getAverageScore()));
    }
}
```