Notice

- Most people infected with COVID-19 feel lethargic for one week, during which they cannot work. If you leave the assignment to the last week but fall victim to the misfortune of catching COVID-19, it is very unlikely you can finish it. You are thus advised to plan your work early.

  Please be reminded: We cannot accept late submissions for **any excuse**.

- For each question, write the following information as a comment to the method:

  1) all the Internet resources you've used for solving the problem (even though you didn't explicitly copy from them);
  2) all the students with whom you have discussed this question; and
  3) running time of your method, with the smallest possible function.[1]

- Make your algorithms as efficient as possible, in terms of both time and space, with priority on time. Your scores depend on their efficiency.

- No marks shall be awarded if your submission does not compile.[2]

- Submission procedure (each deviation will result in deduction of 10 points):

  1) Name the .java file as `<class_name>_<your_id>_<your_name>.java`, e.g., `GradeBook_12345678d_TangTszkei.java` if your name is Tang Tszkei and your ID is 12345678d. (You may find Alt-Shift-R helpful if you're using Eclipse.)
  2) Put all your files into a folder with name `A1_<your_id>_<your_name>`.
  3) Create a .zip or .jar file to contain this folder, similar as the distributed file. **Warning: only zip and jar are accepted**.

[1] A frequent question is "Whether I can use methods from Java library?" Most library methods are complicated and difficult to analyze. If you use them, you have to count their steps, which is almost impossible from my experience.

[2] If you've problems, this and this web pages may help, and you're welcome to seek help on the discussion forum (DON'T SHARE YOUR CODE).

1. (50 points) The subject COMP2011 is offered in two sessions. I need two different views of the grade book of COMP2011.

   Session view Displayed session by session. The students are listed in alphabetical order by their IDs in each session.

   Grade view Displayed by grades, and for students with the same grade, students from session 1 are listed before session 2.

Session	1	1	2	1	2
ID	10028	10010	10167	10107	10047
grade	D	B	B	A	B

   For example, if there are five students, as in the table above (each column for a student), they will be shown as

   - [10010, 10028, 10107, 10047, 10167] in the session view, and
   - [10107, 10010, 10167, 10047, 10028] or [10107, 10010, 10047, 10167, 10028] in the grade view.

   Implement the following methods in `GradeBook.java`. The input is in randome order.

```
void sessionView(Student[] a) {}
void gradeView(Student[] a) {}
```

2. (50 points) A car dealer wants to arrange the cars by their features. He ranks k features by their importance: they are given as a boolean array, where a lower-indexed one is more important. If two cars A and B are equivalent in the first i features and differ on the (i+1)st, then the one having the (i+1)st feature is put before the one that does not.

   For example, if Cruise Control is more important than Front Parking Sensors, then the output is

$$A \to C \to D \to B \downarrow$$

   for the following four cars

   - Car A has both Cruise Control and Front Parking Sensors.
   - Car B has neither Cruise Control nor Front Parking Sensors.
   - Car C has Cruise Control but not Front Parking Sensors.
   - Car D has Front Parking Sensors but not Cruise Control.

   Implement the following method in `CarDealer.java`. Both the input and the output are doubly linked lists of n nodes.

```
DList<Car> sort(DList<Car> cars, int  k)
```

   Hint: split the doubly linked list into a sequence of doubly linked lists, then merge them back to a single one. Think carefully about how to maintain this sequence.