Part 4: You will now create an alternate way to access student records, using a *Hash Table* and the Student ID to store, search and retrieve information.  
When a student is added to the BST, the information will also be added to a “hast table” as follows:  
1. Create an array (rosterTable) of students  
2. Each student will be stored in the array by the last 2 digits of IDNumber. For example the student (John Smith S123456) would be stored in rsterTable[56]. This allows a **unique** index in the array to store the student, with an O(1) insert, delete, and search runtime. The hash table does not compare students or sort them, it merely places the information in a location based on the ID rather than the name.  
  
You can just add the rosterArray placement into your main method addStudent, deleteStudent and “search by ID”. (note: up until this point the search by ID was O(n) now it will be O(1)!) You do not need to add a new class for this – just the additional array in main.  
  
For extra credit, you could make the rosterArray an array of ArrayLists, so that if 2 students have the same last 2 digits, they will be stored in the same location.

Part 5: Document the entire project, including overall runtimes for each of the parts. You can create the document as you work on the project, and just submit the finished document separately, or just “Javadoc” to create the documentation as you write the project.

Everything is due August 17, 11:59 PM.