Write a menu driven program that is capable of performing the following operations:

                1- Create a random-access file

                2- Display a random-access file

                3- Retrieve a record

                4- Modify a record

                5- Add a new record

                6- Delete a record

                7- Exit

The random access file will contain student’s academic records having the format below:

|  |  |  |  |
| --- | --- | --- | --- |
| First name | Last name | Student ID | GPA |
| 40 | 40 | 4 | 8 |

where the integers represent the field width in terms of the number of bytes. The students first and last names are at most 20 characters (Unicode characters) long, the ID is an integer, and the GPA is of type double. There are actually two files in this project, a text file and a random access file. You will create a text file containing information for, at least, 10 students (one line per student), and this program will take that information and put it in a random access file.

1- Create a random-access file: It asks the user to enter an input file name (a text file) and an output file name (a random access file).  The program reads the student’s first name, last name, ID, and GPA from the input file, creates a fixed size record as shown above in the memory, and then writes this record to the output file.  It does this as long as the input file is not empty.

2- Display the random-access file: It asks the user to enter a file name.  The program neatly displays the records stored in the file on the screen.  It should display as many records as will fit on a single screen (say 5 records), and then prompt the user with the following:

Return to the main menu (M); Next screen (N); or Display all (A).

3- Retrieve a record: It asks the user to enter a record number and validates the user input. If the user inputs a valid record number, retrieve the record and display it neatly on the screen. Otherwise, display a message indicating the problem and return to the menu.

4- Modify a record: It asks the user for a record number and validates the user input. If the user enters a valid record number, retrieve the record and display it on the screen, allow the user to change zero or more fields of this record, and overwrite the original record in the file with this modified record. If the user enters an invalid record number, display a message indicating the problem and return to the menu.

5- Add a new record: It asks the user to enter data for the new record.  This new record will be appended to the end of the random-access file.

6- Delete a record: It asks the user for the record number and validates the user input. If the user inputs a valid record number, delete the specified record from the random access file using a lazy deletion technique. A lazy deletion doesn’t really delete the record, but makes it in accessible to all the subsequent operations. For example, a deleted record can’t be retrieved, modified, or displayed.

OPERATIONAL SPECIFICATIONS: The program should always return to the main menu after processing the selected item.   The program execution should only be halted if EXIT is selected from the main menu.

SUBMISSION:  Please submit the following in a manila envelope:

            - A hard copy of the source code (well documented).

            - A CD (or DVD) that contains your project folder, input file, and the random access file

            - A hard copy of an input file (create an input file containing at least 10 students)

            - A sample run which tests all the menu options

GRADING:  Documentation: 10%, Program style: 5%, Friendly interface: 10%, Program works: 75%