```
Load the following databases:
<iostream>
<fstream>
<cstring> //for input validation
Using the namespace standard
Void SearchRecord // Function prototype for searching a record to the database

Initialize the following  global constants for array size: name_size, high_size, initials_size, plays_size, revenue_size with their respective size requirements.

Struct <player_record>
    Initialize a char array called name of size name_size
    Initialize an integer array called highscore of size high_size
    Initialize a second char array called initials of size initials_size
    Initialize a second integer array called plays of size plays_size
    Initialize a float array called revenue of size revenue_size


In the Main function
{

    Create String objects to hold the filenames
    Create fstream objects that can store the following information


        Print prompt “Input the database filename”
        Take the filename input from the user

        Print Prompt “input the batch filename”
        Take the filename input from the user

        Print Prompt “Input the Data log filename”
        Take the filename input from the user

    Declare and initialize a struct object here called record
    Initialize a character object called prompt // to ask the user if they want to add another record.

    Validate all the filenames using if else statements and write a record to the binary file

    If the database file has been opened successfully,
        Open the batch file to get the information for the record
            Read the first name and store it in the name variable.
```

```
                Get the information regarding initials, plays, and revenue in the same
                manner

                Write the contents of the record structure to the database File called
                freeplay.dat. The format is as follows:
                record.write(reinterpret_cast<char *>(&record),
                sizeof(record1));

                Write Record Addded to the output log file

                Also write the contents of the struct object called record to the log file in
                the following manner:
                Name: <name>
                High Score: <highscore>
                Initials: <initials>
                Plays: <plays>
                Revenue: $ - formatted to 2 decimal places <revenue>



        Close the database file
        Close the log file
        Close the batch file

        Function call InputValidation()
        Function call searchRecord()
        Function call editRecord()
        Function header deleteRecord()

        Return 0

End
}

Function header for InputValidation ()
        Make sure the "name" contains only alphanumeric characters, punctuation, and spaces
        using isdigit, isspace, etc functions.

        Make sure the highscore array can only store between 1 - 9 digits

        Make sure the initials array contains only three alphanumeric/punctuation character
        Check according to the requirements for the plays and revenue arrays as well.

Function header searchRecord ()

        Initialize 2 long variables to shold the file size and the offset amount (position) :
        ong offset;
        long  numBytes;

        Open the batch file and using the seekg function, get the newline character of the first
        line in the batch file.

        Then move to the next line read from the file what needs to be searched in the database
        file.

        Using the tellg() and the seekg() function as follows, search for the word that was
        previously read.
                Using Seekg function to set the starting position to the end of the file
                Set numBytes equal to tellg to find out how many bytes the file = the position of
                seekg.

                Go back to the beginning of the file and input the position from the batch file
                //You should reach what your are searching for.

                Using a for loop, iterate through across the word to check if the intended word
                has been found in the database File.

        If input was read
                Write  <name> FOUND to the log file and rewrite the rest of the record to the log
                file.
        Else
                Write <name> NOT FOUND to the log file

        Close batch file
        Close log file
        Close database file


Function header editRecord ()
        //search for a record in a file using the seekg and tellg functions like in the searchRecord
        function

        Open the batch file and Read the edit that needs to be made into memory.
        Open the database file for both input and output.
        //The ios::in and ios::out file access flags may be joined with the | operator

        Read the database file for what needs to be replace and write the contents of the batch
        file into the database file.

        Open the logfile for output and write the changes made to the database file to the log
        file.

        Close batch file
        Close database file
        Close log file


Function header deleteRecord()
        Open the batch file and Read what needs to be removed into memory.
                Using the seekg and tellg functions, start reading from line 4.

        Using an iterating for loop, copy all the data except the record that needs to deleted into
        a new file called freeplay1.dat
        Then delete the file as follows:

         if( remove( "myfile.txt" ) != 0 )
            perror( "Error deleting file" );
          else
            puts( "File successfully deleted" );

        Rename the new database file (freeplay1.dat to freeplay.dat) as follows:
        int result;
          char oldname[] ="freeplay1.txt";
          char newname[] ="freeplay.txt";
          result= rename( freeplay1, freeplay );
          if ( result == 0 )
            puts ( "File successfully renamed" );
          else
            perror( "Error renaming file" );
        Open the log file for output and write “RECORD DELETED” with the information
        corresponding to the deleted record to the log fole

        Close the logfile
        Close the batchfile
        Close the database file
```