

CS/CE 1337 – PROJECT 1 – Bishop Cidercade Database

Pseudocode Due: 9/3 by 11:59 PM

Core Implementation Due: 9/12 by 11:59 PM

Project Due: 9/21 by 11:59 PM

KEY ITEMS: Key items are marked in red. Failure to include or complete key items will incur additional deductions as noted beside the item.

Submission and Grading:

- The pseudocode will be submitted in eLearning as a Word or PDF document and is not accepted late.
- All project source code will be submitted in zyLabs.
 - Projects submitted after the due date are subject to the late penalties described in the syllabus.
- Programs must compile using gcc 7.3.0 or higher with the following flags enabled
 - -Wall
 - -Wextra
 - -Wuninitialized
 - -pedantic-errors
 - -Wconversion
- **Type your name and netID in the comments at the top of all files submitted. (-5 points)**

Objective:

- Directly manipulate files using advanced file functions
- Utilize string and character functions to perform basic input validation

Problem: Barcades have become very popular venues over the last few years. A local arcade, Bishop Cidercade, having recently opened near downtown Dallas needs a simple database system to be implemented. Due to time constraints, the database for now is a simple flat file database that contains the name, high score (w/ initials) and the total number of plays of each of their games. The database also contains the total amount each game would earn if not on free play. They have hired you to build a simple interface to interact with the database.

Pseudocode: Your pseudocode should describe the following items

- Main.cpp
 - Detail the step-by-step logic of the main function
 - List other functions you plan to create
 - Determine the parameters
 - Determine the return type
 - Detail the step-by-step logic that the function will perform
- A list of at least 10 test cases you will check during testing (other than the examples below)
 - Specific input is not necessary

- Describe what you are testing
- Examples:
 - Add record to empty file
 - Edit initials for record at end of file
 - Identify name with invalid characters

Zybooks Information:

- Your source file will be named `main.cpp`
- A program template is provided in ZyBooks
 - You will add your code to this template.
 - The code written in the template will take the name of the database from the user and make a local copy of it for manipulation.
 - This is the only way ZyBooks will allow manipulation of local files on the server.
 - The local copy is named `database.dat`
 - Do not make changes to the marked section
- Core implementation has unlimited submissions
 - This will help you make sure the basic actions of your program are working properly in the Zybooks environment
- Final submission is limited to 12 submissions
- White space will be checked to ensure characters are written in the correct place inside the file

Core Implementation:

- Process the sample input file and generate the sample output shown
 - Add record
 - Search record
 - Edit record - update high score
- No invalid input

Details:

- The name of the file you submit will be named **`main.cpp`**
- Start the program by prompting the user for the following information in the order listed
 - Database filename
 - Batch filename
 - These would normally be hardcoded in an application, but zyLabs requires a filename for multiple test files
- The program will perform the following tasks:
 1. Add a record to the database
 2. Search for a record and display it
 3. Edit a record
 4. Delete a record
- **Add a record:** Each record will be added to the end of the file.

- **Search for a record:** The search term will be a word or phrase. Search the name of each game in the file and display the complete record for any game that matches the search term. This match will not be case sensitive. Any part of the name that is the same as the search term is treated as a match.
- **Edit a record:** Given a game name, the program will update the file and confirm the change by displaying the new record on the screen. The following items may be edited:
 1. High score
 2. Initials
 3. Number of plays
 - If number of plays is edited, the revenue will be recalculated
- **Delete a record:** Given a game name, the program will delete the record for that game from the file.
 - One way to do this is to copy all the data except the record to delete into a new file, delete the old file and then rename the new file, however you can implement this any way that you would like. There must not be any blank lines between records
 - You may want to use the *rename* and *remove* functions for this assignment. Both of these functions are located in `<stdio>`.
 - <http://www.cplusplus.com/reference/cstdio/remove/>
 - <http://www.cplusplus.com/reference/cstdio/rename/>
 - You must convert the string holding the filename to a C-string in order for the functions to work (`stringvar.c_str()`).
- All database manipulation must be done in the file (-20 points if not)
 - This will require random file access techniques
- Validate all input in the batch file
 - An invalid command is any command that does not adhere to the format listed below.
 - If invalid input is encountered, ignore the command and proceed with the next command
 - Be prepared for anything in terms of input
- Revenue is based on a quarter per play

Database Format: The database will exist in a file. Prompt the user to enter the filename for the database. Each record will be on a separate line in the file and each line will have a new line character at the end of the line (except for the last line which may or may not have a newline character). Each record in the database will have the following format. Notice that each field will be separated by a comma and a space. All fields except the first have a required size. This is necessary to prevent overwriting the data in other records.

```
<name>, <space><high_score>, <space><initials>, <space><plays>, <space>${<revenue>}
```

- <name> - may be multiple words
- <high_score> - 9 digits – will have leading zeroes if value less than 100,000,000
- <initials> - 3 characters – no white space
- <plays> - 4 digits – will have leading zeroes if value less than 1000
- <revenue> - <4 digits><decimal><2 digits> - will have leading zeroes if first four digits less than 1000

Input:

All input will be file based. A batch file will be used to update the database. Each command in the batch file will be on a separate line in the file and each line will have a new line character at the end of the line (except

for the last line which may or may not have a newline character). The format for each option is listed below. There is a single space between fields.

- Add a record
 - 1<space>"name"<space>high_score<space>initials<space>plays<space>\$revenue
 - The double quotes surround the name so that you know where the end of the name is
 - The name may have spaces
 - The values are not required to have leading zeroes
- Search for a record
 - 2<space><search term>
 - Search term may contain spaces
- Edit a record
 - 3<space>"name"<space><field number><space><new value>
 - <field number>
 - 1 = high score
 - 2 = initials
 - 3 = number of plays
 - The new value is not required to have leading zeroes
 - The double quotes surround the name so that you know where the end of the name is
 - If number of plays changes, recalculate the revenue
- Delete a record
 - 4<space><name>
 - Name may contain spaces
 - Double quotes are not necessary here since there is no data after the name

Input Validation

- Name
 - Contains only alphanumeric characters, punctuation, and spaces
 - Assume double quotes will be around names where indicated in the format.
- High score
 - Contains digits only
 - Minimum 1 character
 - Maximum 9 characters
- Initials
 - Contains exactly 3 characters
 - Characters are alphanumeric or punctuation
- Plays
 - Contains digits only
 - Minimum 1 character
 - Maximum 4 characters
- Revenue
 - Must begin with \$

- Must contain a decimal point
- Must have exactly 2 digits after the decimal point
- Must have at least 1 digit between \$ and the decimal point
- Minimum 1 digit, maximum 4 digits between \$ and the decimal point

Output: Each valid command in the batch file will generate output to the console. After each command output, write a blank line to the console. Do not include any leading zeroes in the output. The output for each command is as follows:

- Add a record
 - RECORD ADDED
 - Name: <name>
 - High Score: <high_score>
 - Initials: <initials>
 - Plays: <plays>
 - Revenue: \$<value> - formatted to 2 decimal places
- Search for a record
 - <name> FOUND or <name> NOT FOUND
 - If found
 - High Score: <high_score>
 - Initials: <initials>
 - Plays: <plays>
 - Revenue: \$<value> - formatted to 2 decimal places
- Edit a record
 - <name> UPDATED
 - UPDATE TO <field> - VALUE <value>
 - Possible fields: high score, initials, plays
 - Name: <name>
 - High Score: <high_score>
 - Initials: <initials>
 - Plays: <plays>
 - Revenue: \$<value> - formatted to 2 decimal places
- Delete a record
 - RECORD DELETED
 - Name: <name>
 - High Score: <high_score>
 - Initials: <initials>
 - Plays: <plays>
 - Revenue: \$<value> - formatted to 2 decimal places