Assignments

Assignment 4 - Returned

Honor Pledge Accepted
Draft - In progress
Submitted
Returned

Assignment Details

Title	Assignment 4
Student	Ramtin Sirjani
Submitted Date	Nov 15, 2022 10:03 PM
Grade	**95.00 (max 100.00)**
History	Nov 15, 2022 10:03 PM EST Ramtin Sirjani (rsirjani) saved draft Nov 15, 2022 10:03 PM EST Ramtin Sirjani (rsirjani) submitted

Instructions

Assignment overview

Congratulations on accepting your new position as lead developer and database admin for the 2211 World Cup. As the resident C expert, you have been assigned the task of designing and implementing a database of all teams (nations) which will be used throughout the World Cup.

Purpose

This assignment will ask you to apply the following concepts from class:

- Basic programming concepts
- Strings
- Structures
- Variable scope

Design

To accomplish this task, each team will be stored as a structure with the following attributes:

- Team code (eg. `0`, `1`, `2`, `3`, `4`, etc.)
  - Each team code must be unique. Only integers 0-31 are acceptable
- Team name (eg. `"Australia"`, `"Cameroon"`, `"Canada"`, `"Argentina"`, `"Belgium"`, etc.)
  - Only team names up to 25 characters in length are acceptable (including the null character - So think of this as 24+1). Each team name must be unique
- Group seeding (eg. `"A1"`, `"B3"`, `"F2"`, etc.)
  - Only groups A-H and seeds 1-4 are acceptable. (So only A1, A2, A3, A4, B1, B2, ..., H3, H4). Each group seeding must be unique
- Primary kit (uniform) colours (eg. "Red", "Orange", "Yellow", "Green", "Blue", "Indigo", and "Violet")
  - Only the values `'R'`, `'O'`, `'Y'`, `'G'`, `'B'`, `'I'`, `'V'` are acceptable.

Your program will then use an array of structures to represent up to 32 possible teams.

Implementation

Your program should continuously prompt the user for one of five possible commands:

1. Insert a new team (using command `i`)
   - Prompt the user for the team code
     - Assume the user will enter one integer
     - This must be unique in your database and cannot conflict with an existing team code. If the number is less than 0 or greater than 31, or if there is a conflict with an existing code, or if the database is full, tell the user the error. The user can try again or you can return the user to the main prompt
   - Prompt the user for the name of the team
     - Assume the user will enter a string of characters of any length
     - If the team name is longer than the acceptable length, you should accept as many characters as you can and ignore any additional characters. If there is any other issue, tell the user the error. The user can try again or you can return the user to the main prompt
   - Prompt the user for the group seeding of the team
     - Assume the user will enter two characters: a letter representing the group, and a number representing the seeding
     - If the letter is not A-H or if the number is not 1-4, tell the user the error. The user can try again or you can return the user to the main prompt
   - Prompt the user for the area of the primary kit (uniform) colour
     - Assume the user will enter one character value
     - If the character is not in the list `'R'`, `'O'`, `'Y'`, `'G'`, `'B'`, `'I'`, `'V'`, tell the user the error. The user can try again or you can return the user to the main prompt
2. Search for an team in the database and print it out (using command `s`)
   - Prompt the user for the team code
     - If the team code is found, print out all the values for this team only (see the print command below for more details)
     - If the team code is not found, tell the user the error. The user can try again or you can return the user to the main prompt
3. Update a team in the database (using command `u`)
   - Prompt the user for the team code
     - If the team code is found, prompt the user to update all the values for the team (see the insert command above for details)
     - If the team code is not found, tell the user the error. The user can try again or you can return the user to the main prompt
4. Print the entire list of teams (using command `p`)
   - Print out a table listing all the teams in your database with all the attributes:
     - Team Code
     - Team Name
     - Group Seeding
     - Primary Kit Colour
5. Quit the program (using command `q`)
   - Yes, all data is lost when quitting your program. You do not need to maintain the data across multiple runs.

Other implementation notes

- You are welcome to create any number of helper functions you wish.

- You are welcome to use any C libraries you wish but you should be able to get by with `stdio.h`
- Note that there is no delete function. This is not an easy task when using an array
- You are welcome to use whatever wording you would like for your prompts. It does not have to precisely match the example below.

Sample output

```
$ ./worldCupDB
******************
* 2211 World Cup *
******************

Enter operation code: p
Team Code   Team Name                 Group Seeding            Primary Kit Colour

Enter operation code: i
        Enter team code: 0
        Enter team name: Canada
        Enter group seeding: F2
        Enter the kit colour: R

Enter operation code: p
Team Code   Team Name                 Group Seeding            Primary Kit Colour
0           Canada                    F2                       Red

Enter operation code: i
        Enter team code: 1
        Enter team name: Australia
        Enter group seeding: D2
        Enter the kit colour: Y

Enter operation code: i
        Enter team code: 3
        Enter team name: Cameroon
        Enter group seeding: G4
        Enter the kit colour: G

Enter operation code: i
        Enter team code: 3
Team already exists.

Enter operation code: p
Team Code   Team Name                 Group Seeding            Primary Kit Colour
0           Canada                    F2                       Red
1           Australia                 D2                       Yellow
3           Cameroon                  G4                       Green

Enter operation code: s
        Enter team code: 1
```

```
Team Code   Team Name                  Group Seeding            Primary Kit Colour
1           Australia                  D2                       Yellow

Enter operation code: u
        Enter team code: 1
        Enter team name: Australia
        Enter group seeding: D2
        Enter the kit colour: B

Enter operation code: p
Team Code   Team Name                  Group Seeding            Primary Kit Colour
0           Canada                     F2                       Red
1           Australia                  D2                       Blue
3           Cameroon                   G4                       Green

Enter operation code: q
$
```

Testing and submitting

You should test your program by compiling and running it on Gaul before submitting (This is how the TA will be testing your program). Capture the screen of your testing by using screenshots. Demonstrate your program works with at least two runs using different input.

Place your program into a `251xxxxxx-Assignment4` folder.

Create a single tarball called `251xxxxxx-Assignment4.tar` containing a directory structure that looks like this:

```
251xxxxxx-Assignment4
   |-------- worldCupDB.png
   |-------- worldCupDB.c
```

Download your tarball and upload it to Assignment 4 in OWL.

I need help

Check to see if your question has already been posted in the forums. If not, post your question.

https://owl.uwo.ca/x/yvBGmc

Submitted Attachments

- 250680632-Assignment4.tar (130 KB; Nov 15, 2022 10:03 pm)

Assignment 4 - C Programming (Structures, Strings, Organization)

Organization and Style

Instructions followed. Properly documented and commented. Useful naming conventions. Appropriate spacing.

Inadequate or Incomplete

Did not follow instructions at all. Commenting is inadequate or missing. Naming is consistently inadequate. Spacing and program layout is difficult to follow.

0 Points

Poor

Did not follow many key instructions. Commenting is generally inadequate. Naming is generally inadequate. Spacing and program layout is difficult to follow.

3.75 Points

Fair

Followed most instructions. Commenting is generally okay. Naming is generally okay. Spacing and program layout is generally okay.

7.5 Points

Good

Followed most instructions. Commenting is generally good. Naming is generally good. Spacing and program layout is generally good.

11.25 Points

Exceptional

Followed instructions. Commenting is exceptional. Naming is exceptional. Spacing and program layout is exceptional.

15 Points

Comment for Organization and Style

Done

0 0

Functionality

Program compiles. Program does what is asked. Program is not buggy.

Inadequate or Incomplete

Program does not compile or compiles with many adjustments. Program is very buggy and/or rarely produces correct results. Or, no submission received.

0 Points

Poor

Program does compile or compiles with minimal adjustments. Program is very buggy and/or rarely produces correct results.

21.25 Points

Fair

Program does compile. Program has only few bugs and usually produces correct results.

42.5 Points

Good

Program does compile. Program has no obvious bugs and usually produces correct results.

63.75 Points

Exceptional

Program does compile. Program has no bugs and consistently produces correct results.

85 Points

Comment for Functionality

Done

0 0

Total: **0**

Additional instructor's comments about your submission

Compiled with Err

Back to list