

Final Report: Mini Project – C Program with the CRUD Functionalities

Final report submission due: 11:59PM Tuesday, April 28, 2020

Mini project span: weeks#5 - 8

- This is a 70% individual and 30% collaborative project running through weeks\$5-8; Please be clear with HGU CSEE Standard on team project:
 - Team project is an educational procedure which helps the students to develop major related skills and abilities as well as teamwork, communication skills, and morality through the experience of carrying out a mission as a team.
 - As a member of the team, each student should fulfill their responsibility by diligently completing individual tasks and respecting other members.
 - Students should do their best in team projects as poor performance of one member impacts that whole team and causes unnecessary damage.
 - In case of mutual peer evaluation in a team project, each member should evaluate other students fairly based on their role and contribution. It is cheating to request or to give higher evaluation beyond the actual contribution, especially to a member who has poorly participated.
 - It is cheating if a team uses the whole or part of the product that is not completed by team members except when it is clearly allowed.
- When finished, submit your work to *LMS*.

1. Please fill in the form below.

Project title: Soccer player management program
Abstract (100-word summary of the software outcome) I made soccer player list control program, user can insert soccer player from file or keyboard. And when user want to update some data about player , then user can search player who user want to change , and can modify values of player. User also can delete data of player. And user can download data in program database. User can download txt file with specific format, or just print it out on screen. User can sort the list of data. User can choose sort by name or age and also can choose descending for ascending order. And if user delete some date, then it filled with NULL values, so if user want to move all NULL datas to end of database, then user can defragment. And finally user can get average age of player in the list
URL to the project on GitHub: https://github.com/21300506kihyun/OSS
Question: Did you double check whether your code in the above repository is up to date? Answer (choose one): Yes

2. Initial proposal

I will create a soccer player name list management program. So we will be free to enter, add, delete, change and print the full list of soccer players. This will help football enthusiasts create their own soccer player dictionary.

3. Self-evaluation form

	Corresponding function(s)	Does it completely satisfy the requirements? If not, explain what is incomplete.	References
Create: Create a new data record from a data file	Create_record()	Yes it completely satisfy the requirements	
Create: Create a new record from the standard input	Create_record()	Yes it completely satisfy the requirements	
Read: According to user specified conditions, print a record, multiple or all records to the standard output	Print_all_records() Printf_specific_records()	Yes, user can print all list of player and if user enter player's name then corresponding records printed.	
Read: Read/write the entire data from/to filesystem	Read_record_from_file() Write_to_file()	Yes it completely satisfy the requirements	
Read: Export the entire data in a report format as a .txt file	Export_to_txtfile()	Yes it completely satisfy the requirements	
Update: Update a specific field of a record, multiple or all records	Update_records()	Yes user can choose update all records or specific records. And can update one or multiple or all records	
Delete: Delete a record, multiple or all records	Delete_all_records() Delete_specific_records()	Yes it completely satisfy the requirements	
Extra: Perform a data defragmentation	Defragment()	Yes it completely satisfy the requirements	
Extra: Arrange (sort) the order of data records	Sort_records()	Yes. User can choose sort by name or age and also can choose descending for ascending order.	

Extra:	Average_age()	Can calculate total average age of records	
--------	---------------	--	--

4. Function descriptions

Create_record() :

Create a new data record from a data file or create a new record from the standard input

Print_all_records() :

Print all records to the standard output.

Printf_specific_records() :

According to user specified conditions, print a record.

Read_record_from_file() :

When user input a file name, then read entire data from filesystem.

Write_to_file():

When user input a file name, then write entire data to that file.

Export_to_txtfile():

Export the entire data in a report format as a .txt file.

Update_records():

user can choose update all records or specific records. And can update one or multiple or all records

Delete_all_records():

Delete all records with value NULL

Delete_specific_records():

If user enter player's name then delete that player's records

Defragment():

if user want to move all NULL datas to end of database, then user can defragment.

Sort_records():

User can choose sort by name or age and also can choose descending for ascending order.

Average_age():

Can calculate total average age of records

5. Screenshots

```

*****
Membership management system
*****
1. Add a new membber
2. Read from file
3. Print all members
4. Print members by name
5. write_to_file
6. export_to_txtfile
7. delete_all_record
8. delete_record_by_name
9. update_record
10. sort_record
11. defragment_record
12. average_age_of_player
99. Quit

```

<main list>

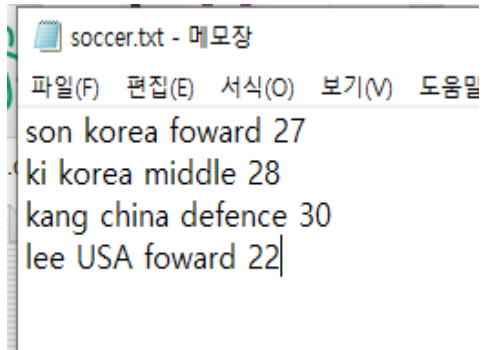
1. Create_records

```

Select a menu> 1
if you want add from file then enter 1 , or keyboard then press 2

```

If user enter number 1, then user can choose create records by file or keyboard.



```

soccer.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
son korea foward 27
ki korea middle 28
kang china defence 30
lee USA foward 22

```

Figure1 soccer.txt

```

if you want add from file then enter 1 , or keyboard then press 2
1
Enter the file name
soccer.txt

```

If user enter number 1, then user can create records by file. If user enter the name of the text file as above, the contents in the text are recorded in the program. At this time, the existing record is remained, and the contents in the file are added to the existing record.

```
Select a menu> 1
if you want add from file then enter 1 , or keyboard then press 2
2
Enter player name, country, position, age : hong afreeca middle 20
*****
```

If user enter number 2, user can receive data through the keyboard. At this time, the existing record is maintained, and the contents in the user input are added to the existing record.

2. Print_all_records()

```
Select a menu> 3
kim korea middle 20
lim japan defence 21
hang china foward 22
```

If user enter number 3 of the menu, all records are displayed..

3. Printf_specific_records() :

```
Select a menu> 4
Enter player name :
kim
kim korea middle 20
*****
```

Next, if user enter number 4, user asked to enter the name of the player. And find the player and print it out. If user search for a player named kim in the above picture, a player named kim is found in the record and printed.

4. Read_record_from_file()

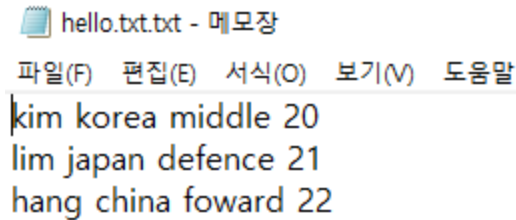
```
Select a menu> 2
Enter the file name
soccer.txt
```

Next, if user enter number 2 in the menu, user asked to enter filename. When user entered, the existing record disappears, and the record is replaced with the data of the file.

5. : Write_to_file():

```
Select a menu> 5
Enter file name :
hello.txt
```

Next, if user enter number 5, user asked to enter the file name and the records are stored in that file



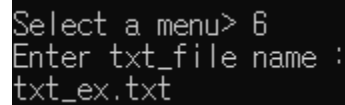
```

hello.txt.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
kim korea middle 20
lim japan defence 21
hang china foward 22

```

<result>

6. Export_to_txtfile():



```

Select a menu> 6
Enter txt_file name :
txt_ex.txt

```

Next, when number 6 was entered, it was saved as a txt file in the specified format



```

txt_ex.txt.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
*****List of Soccer player*****

+++++
Name: kim Country : korea Position : middle Age :20
+++++

+++++
Name: lim Country : japan Position : defence Age :21
+++++

+++++
Name: hang Country : china Position : foward Age :22
+++++

```

<result>

7. Update_records():

If user press 9, user asked update all records or specific records

```
Select a menu> 9
if you want update all records press 1, or update specific records then press 2
1
Enter player name, country, position, age : hong USA middle 30
*****
```

If user enter number 1, user can update all records

```
Select a menu> 3
kim korea middle 20
lim japan defence 21
hang china foward 22
kim korea middle 20
lim japan defence 21
hang china foward 22
*****
```

<before>

```
Select a menu> 3
hong USA middle 30
hong USA middle 30
hong USA middle 30
hong USA middle 30
hong USA middle 30
hong USA middle 30
*****
```

<after update>

```
Select a menu> 9
if you want update all records press 1, or update specific records then press 2
2
Enter the player name
kim
Enter player name, country, position, age : yang germany defence 22
*****
```

If user enter number 2, then user can update specific records

```
Select a menu> 3
kim korea middle 20
lim japan defence 21
hang china foward 22
kim korea middle 20
lim japan defence 21
hang china foward 22
*****
```

<Before>

```
Select a menu> 3
yang germany defence 22
lim japan defence 21
hang china foward 22
yang germany defence 22
lim japan defence 21
hang china foward 22
*****
```

<After>

8. Delete_specific_records():

```
Select a menu> 8
Enter player name :
kim
*****
```

If user enter number 8, user can choose player who want to delete.

```
Select a menu> 3
kim korea middle 20
lim japan defence 21
hang china foward 22
kim korea middle 20
lim japan defence 21
hang china foward 22
*****
```

<before>

```
Select a menu> 3
NULL NULL NULL 0
lim japan defence 21
hang china foward 22
NULL NULL NULL 0
lim japan defence 21
hang china foward 22
*****
```

<After>

9. Delete_all_records() :

If user enter number 7, then user can delete all records.

```
Select a menu> 7
All record deleted
*****
```

10. Defragment():

if user enter number 11, then user can move all NULL datas to end of database

```
Select a menu> 3
NULL NULL NULL 0
lim japan defence 21
hang china foward 22
NULL NULL NULL 0
lim japan defence 21
hang china foward 22
*****
```

<Before>

```
Select a menu> 11
lim japan defence 21
hang china foward 22
lim japan defence 21
hang china foward 22
*****
```

<After>

11. Sort_records():

User can sort the records, and can choose how to sort.

If user enter number 10 user can sort the records.

```
Select a menu> 10
Enter 1 to sort by age , Enter 2 sort by name
1
Enter 3 to ascending order, Enter 4 to descending order
3
*****
```

If user want to sort by age, then enter 1, and then user can also choose ascending or descending.

```
Select a menu> 3
lim japan defence 21
hang china foward 22
lim japan defence 21
hang china foward 22
son korea foward 27
ki korea middle 28
lee USA foward 22
kang china defence 30
*****
```

<Before>

```
Select a menu> 3
lim japan defence 21
lim japan defence 21
hang china foward 22
hang china foward 22
lee USA foward 22
son korea foward 27
ki korea middle 28
kang china defence 30
*****
```

<After>

If user want to sort by name, then enter 2, and then user can also choose ascending or descending.

```
Select a menu> 10
Enter 1 to sort by age , Enter 2 sort by name
2
Enter 5 to ascending order, Enter 6 to descending order
6
*****
```

```
Select a menu> 3
son korea foward 27
lim japan defence 21
lim japan defence 21
lee USA foward 22
ki korea middle 28
kang china defence 30
hang china foward 22
hang china foward 22
*****
```

<Result>

12. Average_age():

If user enter number 12, then user can calculate total average age of records

```
Select a menu> 12
average_age : 24
*****
```