

**In The Name Of God**  
**The most compassionate and merciful**

---

- #1 : any fraud or cheating will negatively affect your score.**
  - #2 : please use ANSI C standard in coding .**  
**(30% of your score stands for standard and clean coding )**
  - #3 : syntax error is equal to zero .**
  - #3 : you should only zip and upload your ".c" files .**
  - #4 : name your zip as " stdNo\_hw2.zip"**
  - #5 : name your .c files as follows : "stdNo\_HW2\_Q?.c"**  
**( ? stands for the question number which is between 1-6 )**
- 

1. Write a program to manage your contacts info, you must use linked list data structure for your phone book:
  - a. Write a structure called contact to keep a person's first name, last name and phone number, as strings.
  - b. Write a structure as linked list's node, which keeps a contact as data.
  - c. Write a function to make and insert a new contact in your phone book if there's no contact with same phone number, you have to keep your contacts sorted by last name, without using sort functions.

Declaration:

```
node insert (node phone_book, char * first_name, char * last_name, char *phone_number); // first  
argument(phone_book) is the first node of linked list, if it was null then your phone book is empty.  
// return value must be the first node too
```

- d. Write a function to remove a contact which has the given phone number (if there is such a contact).

Declaration:

```
node delete (node phone_book, char * phone_number);
```

- e. Write a function to find and print contacts which their last name parameter begins with a specific string.

Declaration:

```
void find (char * prefix);  
void find (char * prefix); // for example if we have ali Mohammadi 111111, farzin Mohammadi 22222, faraz tahmasb  
333333, amir mohammadkhani 44444, ali yazdanpanah 555555 , then the result for "moh" will be :  
ali Mohammadi 11111  
farzin Mohammadi 22222 amir mohammadkhani 33333
```

- f. Write a function to choose and edit contacts which we have chosen with the given number.(hint : use the
- Declaration:

```
node insert (node phone_book, char * phone_number_old, char *phone_number_new);
```

2. Design a simple library management using structs .Create a structure containing book information like id number, name of author, book title and flag to know whether book is available or not.(Linked List is your friend!)  
Create a menu in which the following can be done.
  - 1 - Display book information
  - 2 - Add a new book
  - 3 - Display all the books in the library of a particular author
  - 4 - Display the number of books of a particular title
  - 5 - Display the total number of books in the library
  - 6 - Issue a book

- 
3. Write a program that gets two numbers as char[ ], represents them with linked-lists, and adds them together and prints the result as a char[ ] (You have to create linked-lists for the numbers, add the linked-lists together and create a linked-list for storing the result. You shouldn't add the numbers directly).  
Example:

```
Input:
332
127
Output:
First list:      2 -> 3 -> 3
Second list:     7 -> 2 -> 1
Result list:     9 -> 5 -> 4
Result number:   459

Input:
127
14
Output:
First list:      7 -> 2 -> 1
Second list:     4 -> 1
Result list:     1 -> 4 -> 1
Result number:   141

Input:
22222221111111
44444443333333
Output:
First list:      1 -> 1 -> 1 -> 1 -> 1 -> 1 -> 1 -> 2 -> 2 -> 2 -> 2 -> 2 -> 2 -> 2
Second list:     3 -> 3 -> 3 -> 3 -> 3 -> 3 -> 3 -> 4 -> 4 -> 4 -> 4 -> 4 -> 4 -> 4
Result list:     4 -> 4 -> 4 -> 4 -> 4 -> 4 -> 4 -> 6 -> 6 -> 6 -> 6 -> 6 -> 6 -> 6
Result number:   66666664444444
```

---

4-

برنامه که قرار است بنویسید به این صورت است که ما یک سیستم آموزشی نسبتاً جامع داریم . هر دانشجو دارای اطلاعاتی نظیر شماره دانشجویی ، نام و نام خانوادگی و تعداد نامعینی درس است . هر درس حاوی تعداد نامعینی نمره برای هر دانشجو است. هر درس ضرب مخصوص خود را دارد و هر درس اسم و کد مشخصی دارد.  
در این سیستم ما قرار است در ابتدا منوی به کاربر نشان دهیم بصورت زیر :

- 1 - add student
- 2-search in students by StdNo
- 3-search by name
- 4-sort them by their avg score
- 5-sort them by their name
- 6-delete a student record from the system
- 7-edit a record by giving student number ( only editing stdNo or name and family name )

8-edit student's courses

9-Exit

توضیحی در مورد گزینه 8 :

وقتی کاربر این گزینه را انتخاب میکند ابتدا باید شماره دانشجویی مورد نظرش را وارد کند. پس از اینکه شماره دانشجویی وارد شد تمامی درس هایی که آن دانشجو اخذ کرده است به همراه کد هر درس باید نشان داده شود. سپس کاربر با وارد کردن کد درس و یا اسم درس ( یکی از این دو مورد به انتخاب خودتان ) میتوانید ضریب درس را ادیت کنید و یا به کل آن درس را از لیست دروس دانشجو به همراه تمامی نمراتش حذف کنید . ( هندل کردن این موضوع را میتوانید با گرفتن کدی از کاربر هندل کنید . به فرض مثال کاربر پس از وارد کردن کد درس اگر 0 را وارد کرد پنی درس را حذف کند و اگر 1 را وارد کرد یعنی میخواهد اسم درس را ادیت کند و سیستم باید با چاپ پیامی منتظر گرفتن اسم جدید باشد . ) فقط در جریان باشید که اگر کاربر اسم به درس را از لیست دروس دانشجو حذف کرد ، تمامی پوینتر ها و فضاهایی که گرفته شده بودند باید free شوند .

اگر کاربر اسم درسی را از لیست دروس دانشجو حذف کرد و سپس اقدام به سورت کردن دانشجویان بر اساس نمره کرد، باید میانگین نمرات هر دانشجو را از اول حساب کنید.

توجه کنید که در این همورک فقط و فقط میتوانید از لینک لیست استفاده کنید . از آرایه دو بعدی و یا آرایه های پوینتری استفاده نکنید . فرضتان بر این باشد که هر دانشجو یک المان از یک لینک لیست است که در خود لینک لیستی از دروس دارد و هر درس نیز لینک لیستی از نمرات دارد . (

توجه کنید که به دلیل سنگین بودن گزینه 8 ، اگر هر 7 گزینه دیگر را کامل و درست بنویسید با رعایت خوانایی و نامگذاری درست متغیر ها و استفاده از کامنت و ... در صورتی که گزینه 8 را کامل پیاده سازی نکنید نمره ای از شما کسر نخواهد شد .

Ex :

Welcome.

>>

1 - Add student

2-search in students by Student Number

3-Search by name

4-sort them by their avg score

5-sort them by their name

6-delete a student record from the system

7-edit a record by giving student number

8 - edit student's courses

9-Exit

What is your Choice ? :

>> 8

>> Ok give me a student number : 9732269

>> Ali Behruzi :

Physics {1}

Java {105}

Chemistry {95}

POP { 80 }

Math {121 }

What Course do you wish to edit ?

>> 95 0

>> Ok The Course Has been deleted from 9732269 courses.

Welcome.

>>

1 - Add student

2-search in students by Student Number

3-Search by name

4-sort them by their avg score

5-sort them by their name

6-delete a student record from the system

7-edit a record by giving student number  
8 - edit student's courses  
9- Exit

What is your Choice ? : 1

>> Std Name : Mohammad  
>> Std Lastname : Mohammadian  
>> StdNumber : 9712345  
>> How Many Courses ? : 2  
>> Course 1 Name : POP  
>> Course 1 code : 80  
>> Course 1 Coefficient : 3  
>> Course 1 Scores : 12 14 16 18  
>> Course 2 Name : Math  
>> Course 2 code : 121  
>> Course 2 Coefficient : 2  
>> Course 2 Scores : 10  
>> Student Added .

Welcome.

>>

1 - Add student  
2-Search in students by Student Number  
3-Search by name  
4-Sort them by their avg score  
5-Sort them by their name  
6-Delete a student record from the system  
7-Edit a record by giving student number  
8- Edit student's courses  
9- Exit

What is your Choice ? : 4

Ok !

1 - Soroush Samani :  
Math 121  
Islamic Revolution  
Data Structure  
Avg : 19

2 - Niloofar Zare  
POP  
Fluid Mechanics  
Avg : 18.75

3 - Ali Behruzi  
POP  
Math  
Avg : 12.7

Welcome.

>>

1 - Add student  
2-Search in students by Student Number

```
3-Search by name
4-Sort them by their avg score
5-Sort them by their name
6-Delete a student record from the system
7-Edit a record by giving student number
8- Edit student's courses
9- Exit
```

What is your Choice ? : 9

>> Have a Nice day !

---

5-فرض کنید در اتاقی هستیم میخواهیم n دانش آموز را دور میز دایره ای بنشانیم

[CreateTable](#)

ساخت میز یک عدد میگیرد که ظرفیت میز را مشخص میکند و میزی جدید درست میکند

[AddStudent](#)

تابع افزودن دانش آموز به میز صندلی اضافه میکند تا آن دانش آموز که داری اسم و فامیل مشخصی است روی صندلی بنشیند. اگر خواستیم دانش آموزی اضافه کنیم و صندلی ای وجود نداشت(ظرفیت میز تکمیل شده بود)دانش آموز جدید جایگزین دانش آموزی میشود که اولین نفر نشسته

[GetExam](#)

این تابع باید دانش آموزان را به صورت یکی درمیان دور میز قرار دهد تا امکان تقلب در امتحان را کاهش دهد!!!  
توجه کنید که ممکن است در بعضی اوقات ظرفیت میز به اندازه ی کافی برای یکی درمیان نشستن دانش آموزان نباشد پس باید میز جدیدی ساخته شود و دانش آموزان به میز جدید (یکی درمیان بر همان ترتیب قبلی)اضافه شوند.

[AbsentPresence](#)

تابع حضور و غیاب باید اسم و فامیلی افرادی که دور میز نشستند را به ترتیب اضافه شدنشان نشان دهد.  
دقت کنید که این تابع میتواند بعد از تابع GetExam نیز صدا زده شود

در پیاده سازی توابع محدودیتی از لحاظ تایپ ورودی و خروجی نیست.تنها محدودیت این است که باید. با linked list پیاده سازی شود.