بنام خدا

تکاليف نهايي درس برنامه‌سازي کامپيوتر

تا کنون مسائل کوچک مختلفي را بکمک زبان C پياده‌سازي نموده‌ايد. در اين تکليف، درگير موضوع کاملتري مي‌شويد. سعي شده مسئله‌اي با ابعاد بزرگتر از انواع قبل مطرح شود. پيشنهاد مي‌شود دانشجويان به ترتيب ذيل عمل نمايند :

* درک و تجسم مسئله
* تقسيم مسئله به بخشهاي کوچکتر (زيربخشها)
* پياده‌سازي زيربخشها و آزمايش آنها
* ترکيب زيربخشها در سطوح بالاتر و آزمون آنها تا رسيدن به بالاترين بخش
* آزمون نهايي آن

انتظار دارد پاسخ مسائل خواسته شده را با در نظر گرفتن نکات زير در مهلت

**سه‌شنبه هشتم تيرماه 1389 ساعت 24 به صورت الکترونيکي**

**و**

**چهارشنبه نهم تيرماه 1389 از ساعت 9 الي 14 به صورت شفاهي در محل آزمايشگاه کنترل**

تحويل دهيد :

* نسخة الکترونيکي پاسخ به آدرسهاي

mshora@yahoo.com ، elnaz.shafaei@yahoo.com و s.hemmatan@yahoo.com

ارسال گردد. در صورت دريافت صحيح فايل از اين طريق، پاسخي به ايميل دانشجو ارسال مي‌گردد. در صورت عدم دريافت پاسخ ظرف يک روز، احتمالاً در ارسال ايميل اشکالي وجود داشته که توصيه مي‌‌شود پيگيري نماييد.

* در عنوان (subject) ايميل ارسالي **حتماً** (نام و) نام خانوادگي، موضوع ايميل (تکليف شمارة x درس برنامه‌سازي) را قيد نماييد. انتظار دارد دانشجويان به اين موضوع دقت کنند چرا که احتمال دارد ايميلهاي داراي فيلد subject نامشخص يا نامربوط اساساً باز نشده و حذف شوند.
* شباهت برنامه‌هاي دانشجويان موجب کسر/حذف نمرة تکليف مي‌شود. در صورتي که با يکديگر مشورت مي‌کنيد، سعي کنيد (مثلاً در انتهاي کار( برنامه را يکبار ديگر با فکر خودتان بازنويسي نماييد. معمولاً برنامه‌هاي حاصل از اين بازنويسي در نقاط خاصي تفاوت خواهند داشت که ويژگيهاي فکري هر دانشجو را نشان مي‌دهد.
* اکيداً توصيه مي‌شود که برنامه‌هاي نوشته شده‌تان را بر روي کامپيوتر آزمايش نماييد و از صحت عملکرد آن با آزمونهاي مناسبي که اعمال مي‌نماييد مطمئن شويد.
* در پاسخها حتماً اين موارد را رعايت نماييد :
	+ توضيحات اول برنامه مطابق آنچه در کلاس مطرح شده
	+ توضيح براي بخشهاي برنامه (توضيح خط به خط يا بهتر از آن توضيح براي بخشهاي خاص : بخشي از يک خط و مجموعة چند خط که يک فرآيند از برنامه را تشکيل مي‌دهند).
	+ توضيح مناسب موجب خواناتر و قابل درک‌تر شدن برنامة شما مي‌شود، که نمرة بهتري دريافت مي‌کنيد
* در تحويل شفاهي، دانشجو بايد بتواند
	+ برنامة نوشته شده‌اش را توضيح داده، کامپايل و اجرا نمايد.
	+ به سؤالهايي که از وي در مورد برنامه مي‌شود پاسخ مناسب بدهد
	+ توانايي اعمال تغييرات جزئي در برنامة نوشته شده، که در جلسة ارائة شفاهي از او خواسته مي‌شود، و توضيحات مرتبط با آن را داشته باشد

**مواردي که در اين تکليفها تحويل مي‌شوند شامل :**

* + - **اصل برنامه با توضيحات کافي : تنها فايل با پسوند .cpp قابل قبول است**
		- **نمونة فايل(هاي) اولية ايجاد شده، که بتواند توسط برنامة در حال اجرا (در صورت کامپايل و اجرا شدن فايل فوق) بخوبي مورد استفاده قرار گيرد**
		- **ارائة فايل word و امثال آن انتخابي است**
		- **مجموعة فايلها در يک فايل zip شده و ارسال گردد**

تکليف دوم (4 نمره)

براي تکليف دوم 3 موضوع (مسئله) در نظر گرفته شده است که دانشجويان تکليف تخصيص داده شده را مطابق جدول زير انجام داده و به صورت الکترونيکي ارسال مي‌نمايند. **توجه شود که گروه‌بندي مجدداً انجام شده و مطابق تکليف قبلي نيست.**



تکليف 2 – شمارة 1 - هدف اين تکليف تدوين برنامه‌اي است که بکمک آن، اطلاعات فردي در يک پايگاه داده قابل ذخيره و دسترسي باشد. اين اطلاعات شامل : نام، نام خانوادگي، شمارة تلفن و آدرس ايميل است. براي اين منظور اطلاعات افراد بر روي فايلي ذخيره مي‌شود. اين اطلاعات در قالب ساختار (structure) - با بخشهاي ذيل - بطور متوالي در فايل ذخيره شده‌اند. آخرين عضو ذخيره شده در فايل NULL است يعني همة فيلدهاي آن صفر هستند. از اين طريق رسيدن به انتهاي فايل تشخيص داده مي‌شود.

 struct person\_data {

 char name[31];

 char family\_name[41];

 unsigned long int phone\_number;

 char email[31];

 };

هدف :

* برنامه از کاربر خواهد خواست که يکي از روندهاي زير را انتخاب و دنبال کند
	+ دريافت اطلاعات فردي و افزودن اين اطلاعات به ليست موجود
	+ جستجو در بين اطلاعات موجود
	+ ليست گرفتن از اطلاعات موجود
* در حالت دريافت اطلاعات، در يک فرآيند تکراري (که به شکل مناسبي مي‌تواند قطع شود) اطلاعات افراد به شکل مناسبي از کاربر دريافت مي‌شود. کاربر مي‌تواند برخي فيلدها (نه همة فيلدها) را خالي بگذارد.
* در حالت جستجو :
	+ انتخاب براي جستجو در نام و نام خانوادگي، شماره تلفن و ايميل در نظر گرفته شود
	+ تمامي رخدادهاي ممکن جستجو شوند. مثلاٌ اگر ترکيب مورد جستجو “abc” وارد شود، کلية رخدادهاي اين ترکيب در بين فيلد مورد نظر (ابتداي – انتها - وسط) پيدا شده و در ليست چاپي نمايش داده شوند
* در حالت ليست گرفتن از اطلاعات موجود، کلية اطلاعات به صورت مرتب شده بر حسب يکي از فيلدها (قابل انتخاب) ليست شوند.

تکليف 2 – شمارة 2 – دفترچة حسابهاي بانکي – در نظر بگيريد که شخصي داراي چند حساب بانکي (يا کارت بانکي) باشد. در اين برنامه مي‌خواهيم عملکرد يا گردش حساب شخص قابل ذخيره، مشاهده و ليست‌گيري باشد.

* تا 5 حساب مي‌توان تعريف نمود (مطابق enum زير)

enum transactions {

 balance,

 debit,

 credit

 };

enum accounts {

 melli,

 mellat,

 saman,

 sepah,

 sina

 };

struct date {

 char month;

 char day;

 char year;

 };

struct tx\_data {

 enum accounts account ;

 long int amount;

 struct date time;

 };

* در فايلي که تعريف مي‌شود، عمليات بصورت مرتب و مطابق ساختار tx\_data ذخيره مي‌شوند
* عمليات بر 3 گونه هستند :
	+ اعلام مبلغ باقيمانده (balance) در حساب (نوع حساب و زمان نيز مشخص مي‌شود)
		- طبعاً اولين عملکرد هر حساب در فايل بايد اعلام باقيمانده از دورة قبل (مثلاً سال قبل که در يک فايل ديگر ممکن است ذخيره شده) باشد
		- اگر اولين عملکرد حساب اعلام باقيماندة حساب نبود، باقيماندة حساب «صفر» در نظر گرفته مي‌شود
	+ ورود مبلغ به حساب شخص : «بستانکار يا credit» (نوع حساب و زمان نيز مشخص مي‌شود)
	+ برداشت از حساب شخص : «بدهکار يا debit» (نوع حساب و زمان نيز مشخص مي‌شود)
* آخرين مقدار موجود در فايل (از جنس ساختار tx\_data) NULL خوهد بود (نشان دهندة انتهاي فايل)
* عمليات کاربر چنين هستند :
	+ اعلام (افزودن) عملکرد شامل :
		- حساب مورد نظر
		- نوع عملکرد (اعلام باقيمانده، واريز ِيا برداشت)
		- مقدار عملکرد
		- زمان
	+ گرفتن ليست عملکرد در محدودة تاريخ مشخص مطابق قالب زير :
		- از يک حساب
		- از مجموعة حسابها (جمع دارايي بانکي)

Transactions from 89/1/25 to 89/4/12 for total account

account balance at 89/1/25 : 123,029 rials

date account credit debit balance

89/1/26 melli 12,500 135,529

89/1/27 mellat 13,892 121,637

89/2/06 melli 79,150 42,487

89/2/16 sepah 3,292,956 3,335,443

89/3/20 saman 238,902 3,096,541

89/4/3 sina 87,312 3,183,853

89/4/4 mellat 237,887 2,945,966

تکليف 2 – شمارة 3 – تقويم – در اين برنامه ايام و قرارهاي ثبت شده در يک سال شمسي (مثلاً 1389) مديريت و مشاهده مي‌شوند. فرض بر اين است که :

* تعطيلات بر دو قسم‌اند
	+ جمعه‌ها. بصورت پيش‌فرض کلية جمعه‌ها تعطيل‌اند
	+ ايام تعطيل رسمي بجز جمعه‌ها. اطلاعات اين ايام با قاعدة از پيش تعيين شده از طريق فايلي در اختيار قرار مي‌گيرد.
		- محتويات فايل عناصري از جنس ساختار زير هستند که بصورت متوالي در فايل قرار مي‌گيرند

struct off\_dates {

 char month;

 char day;

 };

* + - مقادير ابتدا و انتهاي فايل مقادير استثنايي دارند
			* فايل به مقدار NULL (هر دو مقدار صفر) ختم مي‌شود
			* از آنجا که موقعيت روز ابتداي سال در هفته (اول فروردين) براي مشخص شدن ايام هفتگي تقويم لازم است، در اولين مقدار در فايل، مقدار month صفر و مقدار day مقداري بين 1 تا 7 (بترتيب متناظر با روزهاي شنبه تا جمعه) قرار داده مي‌شود
		- در فايل ديگري قرارها مطابق ساختار زير ذخيره مي‌شوند :

struct rvs {

 char month;

 char day;

 char hour;

 char appt[101];

 };

* + - * سه مقدار اول روز و ساعت و مقدار چهارم شرح قرار را مشخص مي‌کنند
			* آخرين مقدار در فايل NULL است
* عمليات کاربر شامل :
	+ تعيين (افزودن) قرار ملاقاتها
	+ مشاهدة ليست قرار ملاقاتها (با قالب زير)

appointments from 89/1/25 to 89/4/12

89/1/26 monday coffee shop

89/1/27 thr park

89/2/06 sun English exam

89/2/16 fri hiking

89/3/20 wed pool

89/4/3 mon terminal

89/4/4 tue ceremony

* + مشاهدة تقويم (در قالب زير، روزهاي تعطيل درون کروشه نوشته مي‌شوند)
		- مربوط به ماه مشخص

 sat sun mon tue wed thr fri

ordibehesht

 1 2 [3]

 4 5 6 7 8 9 [10]

 11 12 13 14 15 16 [17]

 18 19 20 21 22 23 [24]

 25 26 [27] 28 29 30 [31]

* + - در محدودة مشخص (مثلاً از تاريخ 23/1/89 تا تاريخ 21/4/89)

 sat sun mon tue wed thr fri

farvardin

 23 24 25 26 [27]

 28 29 30 31

ordibehesht

 1 2 [3]

 4 5 6 7 8 9 [10]

 11 12 13 14 15 16 [17]

 18 19 20 21 22 23 [24]

 25 26 [27] 28 29 30 [31]

khordad

 1 2 3 4 5 6 [7]

 8 9 10 11 12 13 [14]

 [15] 16 17 18 19 20 [21]

 22 23 24 25 26 27 [28]

 29 30 31

tir

 1 2 3 [4]

 [5] 6 7 8 9 10 [11]

 12 13 14 15 16 17 [18]

 [19] 20 21

تکليف سوم – (4 نمره)

هدف اين تکليف تدوين برنامه‌اي است که بکمک آن تعيين گرايش دانشجويان (مثلاٌ برق) صورت گيرد. براي اين منظور اطلاعات دانشجويان بر روي فايلي در اختيار قرار مي‌گيرد. اين اطلاعات در قالب ساختار (structure) با بخشهاي ذيل بطور متوالي در فايل ذخيره شده‌اند.

 enum major {

 notassigned,

 communications,

 control,

 electronics,

 power

 };

 struct st\_data {

 unsigned long int id\_number;

 float total\_average;

 float major\_average[4];

 enum major pref\_major[4];

 enum major assigned\_major;

 };

* در enum تعريف شده، حالت اول به معني «هنوز تعيين گرايش نشده» است و چهار حالت بعدي گرايشهاي ممکن هستند. در ابتداي ورود اطلاعات، براي دانشجويان حالت notassigned در نظر گرفته مي‌شود و پس از تعيين گرايش، گرايشهاي تعيين شده به دانشجويان نسبت داده مي‌شود.
* در ساختار تعريف شده، بخش اول شمارة دانشجويي، بخش دوم معدل کل، بخش سوم آرايه‌اي از چهار معدل وزني از دروس انتخابي ويژة گرايشها با ترتيب مشابه enum فوق، بخش پنجم آرايه‌اي از گرايشهاي انتخابي دانشجو بترتيب اولويت و بخش ششم گرايش نهايي دانشجو (که توسط برنامه تعيين مي‌شود) هستند.

هدف : تعيين گرايش بر اساس معيارهاي ذيل است :

* اولويت انتخاب گرايش با فرد داراي معدل وزني بالاتر در گرايش مربوطه است. در شرايط مساوي، معدل کل نيز در نظر گرفته شود.
* حد تخصيص در هر گرايش به ميزان 25/1 برابر سهم پاية گرايش است. در اينجا براي 4 گرايش، سهم پاية هر گرايش 25% کل دانشجويان (ورودي در يک سال) در نظر گرفته شود. مثلاٌ اگر تعداد اطلاعات وارد شده براي 40 نفر باشد، ظرفيت هر گرايش حداکثر 12 نفر خواهد بود.
* تخصيص گرايش بر اساس اولويتهاي انتخابي دانشجو صورت گيرد. در صورت پر شدن ظرفيت يک گرايش (انتخابي)، اولويت بعدي آزمايش شود.

برنامة نهايي شامل سه قسمت خواهد بود که هنگام شروع برنامه انتخاب يکي را از کاربر مي خواهد :

* توسط يک قسمت، ليست دانشجويان ايجاد و در فايلي (آرايه‌اي از ساختار تعيين شدة فوق که به عنصر null (شمارة دانشجويي صفر) ختم شود) ذخيره مي‌گردد. در صورت وجود ليست قبلي، اطلاعات وارد شده در اين قسمت به ليست قبلي افزوده مي‌شود.
* قسمت ديگر، فايل دادة آماده را مورد استفاده قرار داده و مطابق معيارهاي داده شده تعيين گرايش مي‌نمايد. نتيجة تعيين گرايش در فايل موجود تأثير داده مي‌شود.
* همچنين در قسمت سوم مي‌توان ليستهاي مورد نظر (ليست اوليه و ليستهاي تعيين گرايش شده) را به شکل مناسبي مشاهده نمود (بصورت ليست(هاي) مناسبي چاپ نمود).

روند (الگوريتم) تعيين گرايش مي‌تواند چنين باشد که :

* ليست دانشجويان چهار بار براساس گرايشهاي انتخابي اولويت اول مرتب شود :
	+ ليست افرادي که در انتخاب اول خود گرايش مخابرات را انتخاب کرده‌اند
	+ ليست افرادي که در انتخاب اول خود گرايش کنترل را انتخاب کرده‌اند
	+ ليست افرادي که در انتخاب اول خود گرايش الکترونيک را انتخاب کرده‌اند
	+ ليست افرادي که در انتخاب اول خود گرايش قدرت را انتخاب کرده‌اند
* ليستهاي فوق بر اساس معدل وزني بالاتر در گرايش مربوطه مرتب شوند. در شرايط معدل وزني مساوي، معدل کل براي تعيين موقعيت در ليست بکار گرفته شود.
* از ابتداي هر ليست مرتب شده، به افراد انتخابهاي اول تخصيص داده شود. طبعاً يک يا دو گرايش در فرآيند به مرز پر شدن (1/1 برابر سهم پاية گرايش) مي‌رسند. هر فردي که گرايش خود را دريافت مي‌کند ديگر نبايد در مراحل بعدي تعيين گرايش حضور داشته باشد و لذا بايد به نحوي از ليست حذف شود.
* براي ليست باقيمانده، مجدداً بر اساس اولويت دوم ليستهاي جديدي تشکيل شود (مشابه روند اتخاذ شده براي اولويت اول). روند تخصيص گرايش مشابه روند براي اولويت اول ادامه يابد و افرادي که گرايش را دريافت مي‌کنند از ليست حذف شوند.
* روند براي افراد احتمالي باقيمانده در ليست براي اولويتهاي سوم و چهارم تکرار شود تا ديگر فردي در ليست باقي نماند.