

مدیریت دانشگاه

دانشگاه به برنامه‌ای نیاز دارد که در شروع هر ترم کلاس‌های فیزیکی موجود در دانشکده را به ردوس مختلف تخصیص دهد. نکته مهم این است که دروس تخصیص داده شده به کلاس‌ها دارای تداخل نباشند. مثلاً برای درس برنامه نویسی پیشرفته که باید در روزهای شنبه در ساعت 8:30 شروع و به مدت 2 ساعت برگزار گردد و به ویدئوپروژکتور نیاز دارد، اطلاعات زیر وارد می‌شود:

ID: AP001

Name: Advanced Programming

Day: Saturday

Time:

start:

hour: 8

minute: 30

duration:

hour: 2

minute: 0

Video projector: TRUE

Student list:

880025

871150

850190

850222

887951

889260

884151

Teacher: lotfi

و به همین ترتیب برای سایر دروس اطلاعاتی به همین شکل وارد می‌گردد. علاوه بر موارد فوق لازم است که یک فیلد location (محل) نیز برای هر درس در نظر گرفته شود که این فیلد با شماره کلاس تخصیص داده شده به آن درس پر می‌شود. برای اینکار دو حالت در نظر بگیرید، حالتی که فیلد محل توسط برنامه تعیین گردد و حالت دوم این فیلد مستقیماً توسط مسئول آموزش پر شود، اما برنامه موظف است که درستی قابلیت تخصیص یک کلاس به یک درس را چک کند.

علاوه بر دروس رسمی دانشگاه، یک سری دروس فوق برنامه نیز گاهی برگزار می‌گردند (مثلاً کلاس‌های برگزار شده توسط انجمن علمی). این دروس همه‌ی مشخصه‌های دروس رسمی را دارند، تنها تفاوت آن‌ها این است که این دروس موقتی هستند و تا آخر ترم ادامه ندارند. مثلاً ممکن است دوره‌ای در سه جلسه کامل شود اما دوره‌ی دیگری به 10 جلسه نیاز داشته باشد. این پارامتر در هنگام تعریف درس مشخص می‌شود و در تخصیص کلاس به این دروس باید به این پارامتر نیز توجه داشته باشید.

برای هر کلاس، شماره کلاس، ظرفیت کلاس و اینکه دارای ویدئوپروژکتور هست یا خیر و ID درسی که به آن کلاس تخصیص داده می‌شود، نگهداری می‌کنیم.

الف) اطلاعات مربوط به هر درس و کلیه کلاس‌ها را به نحو مناسب از کاربر دریافت نمایید (مثلاً از یک فایل اطلاعات را بخوانید). توابعی نیز در نظر بگیرید که به کمک آن‌ها بتوانید در صورت لزوم اطلاعات مربوطه را اصلاح کنید. کلید تخصیص دستیابی به یک درس ID آن درس و کلید دستیابی به یک کلاس شماره آن کلاس است که برای اصلاح مشخصات هر یک از آن‌ها با استفاده از این کلیدها می‌توانید جستجوی لازم را انجام دهید.

برای راحتی کار می‌توانید هر شماره دانشجویی را یک int در نظر بگیرید.

تاریخ شامل روز و ماه و سال است و ساعت شامل ساعت و دقیقه شروع کلاس و ساعت و دقیقه مدت زمان کلاس است.

ساعت شروع و پایان کلاس می‌تواند هر بازه‌ای از ساعت باشد. زمان ثابت و دقیقی مد نظر نیست.

حداکثر تعداد نفراتی که می‌توانند در یک کلاس حاضر باشند، 40 نفر است.

راهنمایی: برای ساعت و تاریخ کلاس‌های دیگری تعریف کرده و به نحو مناسب آن‌ها را در کلاس مورد نظرتان استفاده نمایید.

ب) تابعی بنویسید که بتواند مدت زمان برگزاری هر دو کلاس درسی را باهم مقایسه نماید. از خروجی این تابع به نحو مناسب در تخصیص درس‌ها به کلاس‌ها استفاده کنید.

ج) تابعی بنویسید که بر اساس زمان شروع کلاس و مدت زمان کلاس، زمان پایان کلاس را برگرداند.

د) تابعی بنویسید که بتواند بررسی کند دو کلاس از لحاظ محل برگزاری، تداخلی با یکدیگر نداشته باشند. مثلاً اگر کلاس 1 در ساعت 8:15 الی 9:45 برگزار می‌گردد و کلاس 2 از ساعت 9:30 الی 10:30 برگزار می‌گردد،

هر دو این کلاس‌ها نمی‌توانند در کلاس 30 برگزار شوند. پیغام‌های مناسب در این رابطه در داخل خود تابع چاپ شوند.

راهنمایی: برای پیاده‌سازی این تابع لازم نیست آن را حتماً به صورت عضو کلاس بنویسید (بهتر است از بحث توابع friend کمک بگیرید). در ضمن از تابعی که زمان پایان هر کلاس را محاسبه می‌کند نیز می‌توانید در این تابع کمک بگیرید.

ه) تابعی بنویسید که بتواند یک گزارش کلی از لیست کلاس‌ها شامل، ID کلاس، نام کلاس، تعداد دانشجویان کلاس، روز و ساعت شروع و پایان کلاس، نام استاد کلاس، محل برگزاری کلاس و دارای ویدئوپروژکتور بودن کلاس درست کرده و در یک فایل txt ذخیره کند. همچنین فایل گزارش باید شامل لیست کامل از دانشجویان و اساتید (بدون تکرار) باشد.

خلاقیت در طراحی فایل گزارش بر عهده دانشجو می‌باشد اما موارد گفته شده باید در گزارش موجود باشند.

در پیاده‌سازی این پروژه نیز نکات زیر را مد نظر قرار دهید:

- شی‌گرایی کامل برنامه
- ارث‌بری (Inheritance)
- چندریختی (Polymorphism)
- بکارگیری الگو (Template)
- تقسیم‌بندی مناسب کد و توزیع منطقی آن در فایل‌های مختلف
- هر دانشجو و هر استاد می‌توانند در هر لحظه در یک درس حضور داشته باشند؛ برنامه باید قادر به کنترل این مورد و پیشگیری از ثبت موارد احتمالاً اشتباه وارد شده باشد.
- ID درس‌ها و شماره کلاس‌ها باید منحصر بفرد باشند. برنامه باید هنگام ورود اطلاعات این مورد را ارزیابی کند و از ثبت ID های تکراری و یا شماره کلاس‌های تکراری جلوگیری کند.
- استفاده مناسب از فایل‌ها برای ورودی/خروجی : استفاده از توابع کتابخانه‌ای مربوط به فایل‌ها در زبان ++C اجباری است. از مباحث فایل C استفاده نکنید.
- برنامه دارای کامنت‌های خوانا و ترجیحاً به زبان انگلیسی باشد.
- برای مرتب‌سازی از الگوریتم Quicksort یا Merge Sort استفاده نمایید.
- برای جستجو از روش جستجوی دودویی یا Binary Search بهره ببرید.

- در این پروژه از پایگاه داده رابطه ای استفاده نکنید.
- در مواردی که به طور دقیق عملکرد مورد نیاز برنامه مشخص نشده است، به صور دلخواه عمل کنید و در مستندات برنامه راجب آن توضیح دهید.
- پروژه باید به صورت یک فایل ZIP حداکثر 20 مگابایتی در Quera آپلود شود. این فایل ZIP شامل پوشه پروژه (که دارای تمامی کدهای پروژه می باشد) و یک فایل PDF از مستندات پروژه باشد.

موارد زیر نمره اضافه دارد:

- استفاده از یک نرم افزار Version Control مانند Git
- داشتن رابط کاربری گرافیکی (GUI)
- استفاده از Visual Studio یا QT