

باسمه تعالی

## پروژه درس برنامه‌نویسی پاسکال

نیمسال دوم ۹۰-۱۳۸۹

**چند نکته:**

لطفاً مطالب این فایل را به دقت مطالعه کنید.  
به اطلاعاتی‌های آتی که روی سایت قرار می‌گیرد توجه کنید.

برای هرگونه رفع اشکال می‌توانید از دوستان، اساتید حل تمرین یا استاد خود کمک بگیرید.  
علاوه بر آن تلاش می‌شود یک کلاس رفع اشکال پروژه برگزار شود که از طریق سایت اعلام خواهد شد.

تاریخ‌های مهم که در متن این فایل به آنها اشاره شده است:

۱۴ تیر: ارسال پیش‌گزارش اول (مطابق آنچه در ادامه گفته شده است)

۲۱ تیر: ارسال پیش‌گزارش دوم (مطابق آنچه در ادامه گفته شده است)

۲۲ تیر: مراجعه به سایت در مورد زمان نهایی و نحوه تحویل پروژه و ارائه آن

## تذکرات مهم

- ✚ پروژه‌ها می‌تواند به صورت یک نفره یا در گروه‌های دو نفره انجام شود (از نظر نمره تفاوتی ندارد).
- همگروهی‌تان را می‌توانید از میان دانشجویان تمام گروه‌های درس پاسکال این ترم انتخاب کنید.
- ✚ برای نوشتن هر کدام از برنامه‌ها، می‌توانید به دلخواه از زبان دلفی یا پاسکال استفاده نمایید.  
(انتخاب زبان برنامه‌نویسی در نمره شما تأثیری نخواهد داشت.)
- ✚ زمان تحویل و ارائه پروژه، احتمالاً در بازه زمانی ۲۷ تا ۳۱ تیر خواهد بود. زمان دقیق آن و نحوه تحویل، متعاقباً اعلام خواهد شد (این‌شاء... تا روز ۲۲ تیر روی سایت قرار می‌گیرد).
- ✚ **حضور هر فرد به هنگام ارائه پروژه در سایت کامپیوتر اجباری است.**
- ✚ برنامه شما باید روی هر کامپیوتری و در هر درایوی که نصب شد به راحتی اجرا شود و نیاز به تغییر Source برنامه، برای تغییر مسیر فایل‌ها نداشته باشد.
- ✚ برنامه شما باید دارای Help باشد (درون برنامه یا در فایل جداگانه).
- ✚ لازم است هر فرد به تنهایی و جداگانه دو گزارش پیشرفت پروژه در تاریخ‌های ۱۴ و ۲۱ تیرماه از طریق آدرس [cw.sharif.edu](http://cw.sharif.edu) تحویل دهد. گزارش پیشرفت شامل موارد زیر است:
  ۱. نام خود و همگروهی‌تان (در صورت همگروهی داشتن باید تا قبل از اولین گزارش، مشخص شود)
  ۲. برنامه نوشته شده تا آن تاریخ (حتی اگر ناقص است یا اجرا نمی‌شود)
  ۳. یک صفحه گزارش تایپ شده به صورت فایل word یا pdf (شرح مختصر فعالیت‌های انجام شده بر روی پروژه و نتایج آن)
- لازم است تمام موارد فوق را به صورت یک فایل فشرده (ZIP یا RAR) به نام شماره دانشجویی خودتان Upload کنید (مثلاً 89123456.zip).
- ✚ در تمام پروژه‌ها، علاوه بر مواردی که ذکر خواهد شد، استفاده از ماوس، استفاده از موزیک و هر مورد اضافه دیگری که به ذهنتان می‌رسد می‌تواند نمره اضافه داشته باشد.

✚ نحوه ارزیابی پروژه شما (از ۱۰۰ نمره):

کدنویسی	۴۰ نمره
ماژولار نوشتن (استفاده مناسب از زیربرنامه‌ها)	۱۶ نمره
خوانایی برنامه	۸ نمره
نوشتن توضیحات مناسب در برنامه	۸ نمره
نام‌گذاری مناسب متغیرها، زیربرنامه‌ها و ...	۸ نمره
عملکرد برنامه	۴۰ نمره
ریز موارد بر اساس کارهایی است که برنامه باید انجام دهد	
اجرای برنامه	۲۰ نمره
ظاهر مناسب (سعی کنید در محیط گرافیک باشد)	۸ نمره
User-friendly بودن برنامه	۶ نمره
پیش‌بینی حرکات کاربر (پیش‌گیری از خطاهای ناخواسته)	۶ نمره

**در روز تحویل و ارائه پروژه، حتماً CD یا Flash Memory حاوی فایل‌های پروژه خود را**

**به همراه داشته باشید (ترجیحاً هم CD و هم Flash).**

## صورت پروژه

در این پروژه شما باید یک برنامه انتخاب واحد بنویسید. سه فایل به نام‌های **Users.dat**، **Courses.dat** و **Scores.dat** در دایرکتوری جاری در اختیار داریم. (لازم نیست این فایل‌ها را ایجاد کنید. هنگام تحویل پروژه، این فایل‌ها در کنار برنامه شما قرار می‌گیرد. سعی می‌شود چند فایل نمونه از قبل در اختیارتان قرار گیرد.)

در فایل **Users** مشخصات دانشجویان آمده است. این مشخصات شامل شماره دانشجویی، رمز عبور به سیستم و رشته تحصیلی دانشجو است. فایل را از نوع رکورد زیر تعریف کرده‌ایم:

**Student = Record**

**StdNO : Longint;**

**Password : String[20];**

**Field : String[20];**

**End;**

در ابتدای برنامه، دانشجو شماره دانشجویی و رمز عبور خود را وارد می‌کند و در صورت درست بودن، وارد سیستم می‌شود.

فایل **Scores** از نوع رکورد زیر است:

**Score = Record**

**StdNO : Longint;**

**Crs\_ID : Word;**

**Unit : Byte;**

**Point : Real;**

**End;**

منظور از Crs\_ID کد درس مربوطه و Unit تعداد واحد آن است. در این فایل، نمرات تمام دانشجویان وجود دارد.

مثلا فایل به صورت زیر است:

StdNO	Crs_ID	Unit	Point
89123456	40151	3	20
89123456	22345	4	16.5
89100000	24789	2	18.3
89999999	40151	3	17
89100000	40151	3	19
89123456	24789	2	12
89100000	22345	4	15.9

هر کاربر پس از ورود به سیستم از طریق منویی باید قادر باشد کارنامه خود را مشاهده کند (لیست نمرات و معدل).

فایل Courses شامل لیست دروس ارائه شده در نیمسال جاری و مشخصات آن دروس می باشد؛ به ترتیب زیر:

**Course = Record**

**Field : String[20];**

**Crs\_Name : String[20];**

**Crs\_ID : Word;**

**Unit : Byte;**

**Pre\_ID : Word;**

**Group : Byte;**

**Day : String[3];**

**Start : Byte;**

**Finish : Byte;**

**Exam : Date;**

**End;**

که نوع داده Date به صورت Year, Month, Day : Byte است.

عبارت Field مشخص می کند این درس مربوط به چه رشته ای است (بعضی دروس مربوط به یک رشته خاص است و برخی دروس پایه Basic هستند). نام و کد درس و تعداد واحد آن با Crs\_ID، Crs\_Name و Unit تعیین می - شود. عبارت Pre\_ID کد درس پیش نیاز این درس را نشان می دهد (فرض کنید هر درس حداکثر یک پیش نیاز دارد؛ صفر یعنی بدون پیش نیاز). شماره گروه درس ارائه شده در Group آمده است و Day معلوم می کند در کدام روز هفته این درس ارائه می شود (مثلا برای Sat برای شنبه و ...). برای سادگی فرض می شود هر درس فقط در یک روز تدریس می شود. متغیرهای Start و Finish نشان می دهد این درس از چه ساعتی تا چه ساعتی در روز تدریس می شود (فرض کنید همه دروس در ساعت های کامل شروع و تمام می شوند؛ یعنی مثلا ۸، ۹، ۱۰، ... و مثلا ساعت ۸:۳۰ نداریم). در نهایت Exam تاریخ امتحان پایان ترم مربوط به درس است. مثلا جدول زیر نشان می - دهد درس ۳ واحدی پاسکال به شماره ۴۰۱۵۱ جزء دروس پایه بوده (رشته Basic) و بدون پیش نیاز در گروه ۱، ساعت ۸ الی ۱۱ روزهای چهارشنبه تشکیل می شود و تاریخ امتحان آن ۹۰/۳/۲۱ است:

Field	Crs_ID	Crs_Name	Unit	Pre_ID	Group	Day	Start	Finish	Exam
Basic	40151	Pascal	3	0	1	Wed	8	11	90   3   21

هر کاربر پس از ورود به سیستم باید بتواند از طریق منویی که در اختیار دارد، انتخاب واحد کند. بدین ترتیب که ابتدا فهرست دروس مربوط به آن رشته (به علاوه دروس پایه) نمایش داده شود. ممکن است تعداد دروس زیاد باشد. پس باید فهرست دروس را در چند صفحه نشان دهید (هر پنج درس در یک صفحه، دکمه ای برای رفتن به صفحات بعد و قبل وجود داشته باشد).

کاربر کد درس و شماره گروه را وارد می کند. برنامه شما باید چند مطلب را بررسی کند:

۱- اگر درس انتخابی پیش نیاز دارد، آیا دانشجو پیش نیاز آن درس را پاس کرده یا خیر.

۲- دانشجو دو درس که تاریخ امتحان آنها یکی است را همزمان انتخاب نکند.

۳- دانشجو دو درسی که روز و ساعات آن با هم تلافی دارد را انتخاب نکند.

۴- درس انتخاب شده مربوط به رشته دانشجو باشد یا اینکه درس مربوطه یک درس پایه باشد (Field = Basic)

در صورت وجود مشکل، پیغام خطای مناسب نمایش داده شود. در غیر اینصورت، درس انتخابی به فهرست دروس

نیمسال جاری وی افزوده شود. در پایان کاربر باید بتواند لیست ثبت نام خود را به همراه زمان کلاس و امتحان

مشاهده کند.

سایر امکانات اجباری:

باید منویی در برنامه شما وجود داشته باشد که برای هر کاربر، از میان دروسی که می تواند انتخاب کند، یک

برنامه درسی مناسب پیشنهاد دهد؛ طوری که پیش نیازی و تلافی ها رعایت شود.

### **امکانات اجباری برای گروه های دو نفره (این بخش، برای گروه های تک نفره هیچ امتیازی ندارد)**

• کاربری با نام 1 و رمز admin وجود داشته باشد که پس از ورود به سیستم بتواند یک کاربر جدید (با

شماره دانشجویی و رمز عبور جدید) به سیستم اضافه کند یا یک کاربر را حذف کند.

• کاربر admin بتواند یک درس جدید را به مجموعه دروس اضافه کند.

## امکانات اضافی برنامه (اختیاری)

- هر کاربر پس از ورود به سیستم بتواند رمز عبور خود را تغییر دهد.
- کاربری با نام 2 و رمز teacher وجود داشته باشد که بتواند نمره دروس دانشجویان را وارد کند و این نمرات در فایل Scores ثبت شود.
- ...