



مقدمه

هدف از این تمرین آشنایی شما با رشته‌ها و وکتورها است. در این تمرین شما یک نمونه ساده شده از سامانه انتخاب واحد را طراحی خواهید کرد. علاوه بر اجرای درست برنامه، جدا کردن مسئولیت‌ها به تابع‌های مختلف و تمیزی کد اهمیت زیادی دارند. سعی کنید در پیاده‌سازی تمرین، توابع مختلفی تعریف کنید که هر کدام تنها یک کار انجام می‌دهند.

سامانه انتخاب واحد

دانشجویان در ابتدای هر ترم باید با دروسهای انتخاب واحد در سامانه گلستان (که حالا بعد از یکم تغییرات در طراحی بهش می‌گن سامانه بهستان!) سر و کله بزنند. اما برخلاف عقیده اکثر دانشجویان، نوشتن یک سامانه انتخاب واحد به هیچ وجه کار راحتی نیست. در این بخش قصد داریم یک نمونه خیلی ساده شده از این سامانه را بنویسیم تا با برخی از دشواری‌های آن آشنا شویم.

فهرست دروس ارائه‌شده در ترم آتی

به صورت کلی، فهرست دروس ارائه شده معمولاً در پایگاه‌داده^۱ سامانه ذخیره می‌شوند. در این تمرین برای جلوگیری از افزایش پیچیدگی، تمامی دروس ارائه شده در ترم آتی، از طریق ورودی استاندارد^۲ و به نحوی که در ادامه توضیح داده شده، به برنامه شما داده می‌شود.

دروس گذرانده‌شده

مشابه مورد قبلی، فهرست دروسی که هر دانشجو در ترم‌های گذشته گذرانده به همراه نمره آن‌ها در پایگاه‌داده سامانه ذخیره می‌شود. اما مجدداً در این مورد برای سادگی، این فهرست در ورودی استاندارد به شما داده خواهد شد.

^۱ Database

^۲ Standard Input

اخذ درس در ترم آتی

دانشجویان می‌توانند در این سامانه، از بین دروسی که در ترم آتی ارائه می‌شود، بعضی از دروس را اخذ کنند. اما نکته مهم این است که برخی شروط باید در زمان اخذ این دروس رعایت شود که در ادامه ذکر می‌شوند.

ارائه شدن درس در ترم آتی

دروسی که دانشجو قصد اخذ آن‌ها را دارد، باید در لیست دروس ارائه‌شده در ترم بعد وجود داشته باشند.

رعایت کف واحد

در این بخش کف واحد را به صورت ثابت و برابر با ۱۲ واحد در نظر می‌گیریم. حالت‌های خاص که دانشجو می‌تواند کمتر از ۱۲ واحد اخذ کند (نظیر ترم آخری بودن) در آزمون‌ها وجود نخواهد داشت.

رعایت سقف واحد

در صورتی که معدل کل دانشجو بزرگ‌تر و یا مساوی ۱۷ باشد، می‌تواند حداکثر ۲۴ واحد اخذ کند. اگر معدل کل او بزرگ‌تر یا مساوی ۱۲ و کوچک‌تر از ۱۷ باشد، می‌تواند حداکثر ۲۰ واحد، و در غیر این صورت حداکثر ۱۴ واحد اخذ کند.

برای محاسبه معدل برای n درس، می‌توانید از رابطه زیر استفاده کنید:

$$GPA = \frac{\sum_{i=1}^n g_i \times u_i}{\sum_{i=1}^n u_i}$$

به طوری که g_i نمره درس i -ام و u_i تعداد واحد درس i -ام است. در معدل، تمام دروس، حتی دروسی که دانشجو در آن‌ها مردود شده محاسبه می‌شوند.

عدم اخذ درسی که قبلاً گذرانده شده

دانشجو امکان اخذ درسی که پیش از این ترم آن را با نمره قبولی (بزرگ‌تر و یا مساوی ۱۰) گذرانده را نخواهد داشت.

رعایت پیش‌نیازهای درس

لازمه اخذ هر درسی توسط دانشجو این است که پیش‌نیازهای آن درس در لیست دروس گذرانده دانشجو باشد و آن درس را با نمره قبولی (بزرگ‌تر یا مساوی ۱۰) گذرانده باشد.

ورودی

در اولین خط ورودی عدد m داده می‌شود که تعداد دروس ارائه‌شده در ترم آتی است. سپس، در هر یک از m خط بعدی، اطلاعات یک درس به صورت زیر آمده است:

پیکربندی ورودی برای یک درس ارائه شده در ترم آتی

```
<course_name> <units> <prerequisites>
```

تضمین می‌شود نام تمامی دروسی که در هر بخشی از ورودی آمده است، بدون فاصله خواهد بود. همچنین، دروس پیش‌نیاز که در قسمت `prerequisites` آمده‌اند، نام دروسی هستند که توسط کاما³ (,) از همدیگر جدا شده و شامل فاصله نیز نیستند. همچنین تضمین می‌شود توصیف دروس یکتاست و دو تعریف درس با نام یکسان نخواهیم داشت.

در ادامه، پس از پایان دروس ارائه‌شده در ترم آتی، در یک خط عدد n ($n > 0$) آمده که نشان‌دهنده تعداد دروس اخذ شده در ترم‌های گذشته است و سپس در هر یک از n خط بعدی، یک درس اخذ شده به صورت زیر داده می‌شود:

پیکربندی ورودی برای یک درس اخذ شده در ترم‌های گذشته

```
<course_name> <units> <grade>
```

بدیهی است که نمره بزرگ‌تر و یا مساوی ۱۰ به معنی گذراندن درس و نمره کوچک‌تر از ۱۰ به معنی عدم قبولی آن درس است. دقت کنید که ممکن است دانشجوی یک درس را بیش از یک بار در ترم‌های اخیر اخذ کرده باشد اما حداکثر یک بار آن شامل نمره قبولی خواهد بود. پس از این مورد، عدد k ذکر می‌شود که تعداد دروسی است که دانشجوی قصد دارد آن‌ها را در ترم آتی اخذ کند. سپس، در خط بعدی، k درس که با فاصله از همدیگر جدا شده‌اند، داده می‌شود.

پیکربندی کلی ورودی به صورت زیر است:

پیکربندی ورودی

```
<m>
<course_name_1> <units_1> <prerequisites_1>
<course_name_2> <units_2> <prerequisites_2>
```

³ Comma

```
...  
<course_name_m> <units_m> <prerequisites_m>  
<n>  
<course_name_1> <units_1> <grade_1>  
...  
<course_name_n> <units_n> <grade_n>  
<k>  
<course_name_1> <course_name_2> ... <course_name_k>
```

خروجی

در صورتی که با توجه به شروط ذکر شده، اخذ این دروس ممکن باشد، عبارت OK! چاپ خواهد شد. در صورتی که درسی در ترم آتی ارائه نشود، با پیغام خطای Course Not Offered! مواجه می‌شویم. اگر کف واحد رعایت نشد با پیغام خطای Minimum Units Validation Failed! و اگر سقف واحد رعایت نشود، با پیغام خطای Maximum Units Validation Failed! مواجه می‌شویم. همچنین اگر قصد اخذ درسی که از قبل آن را گذرانده باشیم، با پیغام Course Already Taken! مواجه می‌شویم. در نهایت، اگر دانشجو پیش‌نیازهای یک درس را نگذرانده باشد، با پیغام خطای Prerequisites Not Met! مواجه خواهیم شد.

بیکربندی خروجی

```
OK! | Minimum Units Validation Failed! | Maximum Units Validation  
Failed! | Course Not Offered! | Course Already Taken! |  
Prerequisites Not Met!
```

رنگی بودن خروجی‌ها در این بخش صرفاً برای فهم راحت‌تر بوده و در تمرین لازم است خروجی‌ها را به صورت غیر رنگی چاپ کنید.

تضمین می‌شود حداکثر یک خطا در دروس اخذ شده وجود خواهد داشت و نیازی به بررسی حالتی که بیش از یک خطا وجود داشته باشد را ندارید.

مثال

ورودی نمونه اول

ورودی نمونه اول

```
6  
NetworkSecurity 3 ComputerNetworks
```

DesignOfAlgorithms 3 DataStructures,ProbabilityAndStatistics
OperatingSystems 3 ComputerArchitecture,DataStructures
NetworkLaboratory 1 ComputerNetworks
CyberPhysicalSystems 3 OperatingSystems
ArtificialIntelligence 3 AdvancedProgramming
5
ComputerNetworks 3 15
DataStructures 3 18
ProbabilityAndStatistics 3 17.5
ComputerArchitecture 3 16
AdvancedProgramming 3 19
5
NetworkSecurity DesignOfAlgorithms OperatingSystems
NetworkLaboratory ArtificialIntelligence

خروجی نمونه اول

خروجی نمونه اول

OK!

ورودی نمونه دوم

ورودی نمونه دوم

6
NetworkSecurity 3 ComputerNetworks
DesignOfAlgorithms 3 DataStructures,ProbabilityAndStatistics
OperatingSystems 3 ComputerArchitecture,DataStructures
NetworkLaboratory 1 ComputerNetworks
CyberPhysicalSystems 3 OperatingSystems
ArtificialIntelligence 3 AdvancedProgramming
5
ComputerNetworks 3 15
DataStructures 3 18
ProbabilityAndStatistics 3 17.5

ComputerArchitecture 3 16

AdvancedProgramming 3 19

5

NetworkSecurity DesignOfAlgorithms OperatingSystems

NetworkLaboratory CyberPhysicalSystems

خروجی نمونه دوم

خروجی نمونه دوم

Prerequisites Not Met!

نکات و نحوهٔ تحویل

- کد خود را در قالب یک فایل با نام A1-SID.cpp در صفحهٔ eLearn درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شمارهٔ دانشجویی شما ۸۱۰۱۰۲۰۰۰ باشد، نام پروندهٔ شما باید A1-810102000.cpp باشد که شامل کد شما است.
- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- بخش مهمی از ارزیابی برنامه شما به درستی عملکرد آن می‌پردازد. بنابراین به اندازه کافی ورودی‌های آزمایشی طراحی کنید تا درستی خروجی در حالت‌های مختلف آزموده شود.
- درستی برنامهٔ شما از طریق آزمون‌های خودکار سنجیده می‌شود. به این ترتیب، لازم است خروجی تولید شده از نظر بزرگی و کوچکی حروف، رعایت فاصله‌ها، عدم وجود خروجی‌های اضافه، ... دقیقاً مانند نمونه‌های داده شده باشد. بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff فرمت خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.

نمرات

- تمیزی کد
 - رعایت کردن نام‌گذاری صحیح و انسجام⁴
 - عدم وجود کد تکراری
 - رعایت دندانه‌گذاری⁵
 - استفاده صحیح از متغیرهای ثابت⁶ به جای Magic Value-ها⁷
 - استفاده صحیح از وکتور و string به جای آرایه و char*
 - توابع کوتاه که فقط یک کار را انجام می‌دهند
- درستی کد
 - آزمون‌های خودکار

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

⁴ Consistency

⁵ Indentation

⁶ Constant

⁷ به مقادیر خاصی که در کد استفاده می‌شود و برای عملکرد صحیح کد ضروری است اما دلیل استفاده از آن‌ها مشخص نیست و قابل جایگزین شدن با یک ثابت با اسم مشخص جهت افزایش خوانایی هستند، magic value گفته می‌شود. برای آشنایی بیشتر با این مفهوم می‌توانید [این لینک](#) را مشاهده کنید.