برنامه‌سازی پیشرفته

نیم‌سال دوم ۹۸-۹۷

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

تمرین دو | شیء گرایی | مهلت ارسال: دوشنبه ۱۲ فروردین ۱۳۹۸

به موارد زیر توجه کنید:

- پاسخ تمرین را در سامانه‌ی کوئرا بارگذاری نمایید.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام‌شده است. توصیه می‌شود نوشتن تمرین را به روزهای نهایی موکول نکنید.
- مهلت ارسال پاسخ تمرین تحت هیچ شرایطی تمدید نخواهد شد.
- هم‌کاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد. ارسال پاسخ تمرین‌ها ۴ روز فرصت ارسال با تاخیر و با جریمه‌ی کسر روزانه ٪۲۰ از نمره‌ی سوال در نظر گرفته شده‌است که برای محاسبه‌ی نمره‌ی سوال در نظر گرفته شده است که برای محاسبه‌ی نمره‌ی تمرین در مجموع ۴ روز از تاخیر تمرین‌ها بخشیده می‌شود.
- به ازای هر روز ارسال زودتر پاسخ هر سوال از تمرین، ٪۵ نمره‌ی مثبت تا سقف ۴ روز (حداکثر ٪۲۰) تعلق خواهد گرفت. (فقط در صورت کسب نمره‌ی کامل سوال که شامل نمره‌ی امتیازی نیز می‌شود)
- مبنای درس، اعتماد به پاسخ ارسالی توسط شماست و در نتیجه در صورت مشاهده‌ی هرگونه مشابهت غیرمتعارف بین کدها، برای هر دو طرف تقلب‌دهنده و تقلب‌گیرنده در مرتبه‌ی اول نمره‌ی ۱۰۰- برای آن تمرین و در صورت تکرار، حذف درس صورت خواهد گرفت.
- به دلیل ماهیت تمرین ۲ و محتمل بودن بروز ابهام در صورت سوالات، مثال‌های مختلف در طول زمان تمرین اضافه خواهند شد. سوالات خود را در مورد تمرین در صفحه‌ی کوئرای درس مطرح کنید.
- برای بررسی همه‌جانبه‌ی کدهای شما تست‌کیس‌های متنوع در طول زمان تمرین اضافه خواهند شد در نتیجه احتمال تغییر نمره‌ی کوئرا وجود دارد.

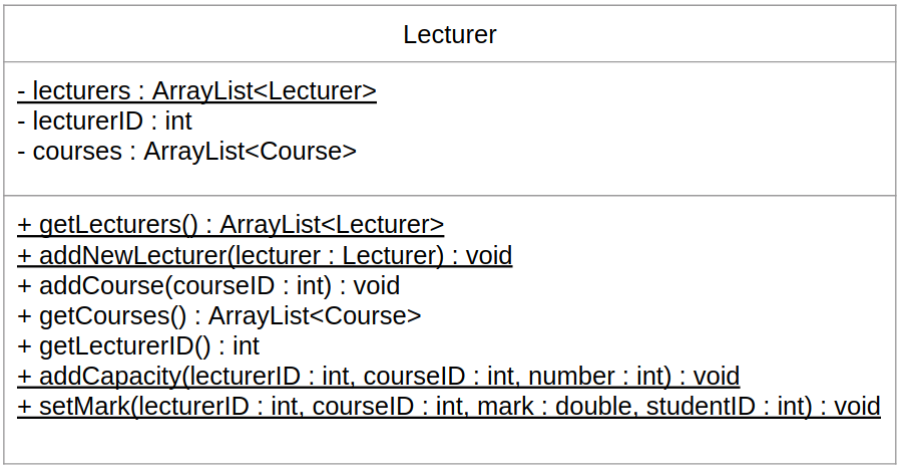
۱ دانشگاه شما دانشگاه نیست! (۲۵ امتیاز)

محمدمهدی بعد از اینکه تکنیک رمزگشایی ساده را یادگرفت، تصمیم گرفت تا انتقام مشکلات انتخاب واحد خود را از مسئولین آموزش بگیرد برای همین با تکنیک‌هایی که بلد بود (!) تلاش کرد تا به سامانه‌ی آموزش نفوذ کند. اما سامانه‌ی آموزش با پیام دانشگاه شما دانشگاه نیست تحمل این نفوذ را نکرد و به طور کامل از دسترس خارج شد. مسئولین آموزش پس از اطلاع از این جریان تصمیم گرفتند به محمدمهدی فرصت جبران بدهند (چرا که اگر او را اخراج می‌کردند میانگین معدل دانشگاه افت می‌کرد!!).
محمدمهدی پس از تفکرات فراوان برای این سامانه، توانست UML ای طراحی کند اما از آنجایی که این حجم از تفکر باعث شد او از درس‌های دیگرش عقب بیفتد از شما به عنوان دوستش کمک خواسته تا این UML را به کد تبدیل کنید و خودش بتواند درس‌های دیگرش را بخواند. او در عوض به شما قول می‌دهد در صورت قبول این زحمت، نمره‌ی سوال ۱ ازتمرین ۲ درس AP را به شکل مخفیانه(!) برای شما وارد سامانه کند.

Course
- courses : ArrayList<Course> - courseID : int - unit : int - capacity : int - numberOfRegisteredStudents : int - score : double
+ addNewCourse(course : Course) : void + getCourses() : ArrayList<Course> + getNumberOfRegisteredStudents() : int + setNumberOfRegisteredStudents(number : int) : void + incrementNumberOfRegisteredStudents() : void + decrementNumberOfRegisteredStudents() : void + getCourseID() : int + addCapacity(number : int) : void + getScore() : double + getUnit() : int + setScore(score : double) : void + getCapacity() : int + showCourse(courseID : int, string : String) : void

Student
- students : ArrayList<Student> - studentID : int - studentCourses : ArrayList<Course> - average : double
+ getAverage() : double + addNewStudent(student : Student) : void + registerCourse(studentID : int, courseID : int) : void + addCourse(course : Course) : void + deleteCourse(StudentID : int, courseID : int) : void + setMark(courseID : int, mark : double) : void + setAverages() : void + getStudents() : ArrayList<Student> + getStudentID() : int + getStudentCourses() : ArrayList<Course> + deleteStudent(studentID : int) : void + showAverage(studentID : int) : void + showRanks() : void + showRanks(courseID : int) : void

نکته: تمامی کلاس‌ها باید مطابق با نمودار UML داده شده پیاده‌سازی شود. در صورتی که نیاز به متد یا کلاس کمکی که در UML ذکر نشده دارید، می‌توانید آن را پیاده‌سازی کنید اما چارچوب کلی کد شما باید مبتنی بر UML داده شده باشد. در غیر این‌صورت محمدمهدی اخراج شده و در نتیجه شما هم نمره‌ی سوال ۱ تمرین ۲ این درس را نمی‌گیرید!



ویژگی‌های سامانه‌ی آموزش

```
addStudent <sID>
```

- دانشجویی با شناسه‌ی sID به سامانه اضافه می‌شود.

```
addLecturer <ltID> <cID1> <cID2> ...
```

- استاد با شناسه‌ی $ltID$ به سامانه اضافه می‌شود. این استاد دروس $cID1$ و $cID2$ و... را ارئه می‌دهد. هم‌چنین ممکن است یک استاد هیچ درسی ارائه نکند یا فقط یک درس ارائه کند.

```
W <cID> <sID>
```

- دانشجوی با شماره‌ی sID درس cID را حذف می‌کند.

```
<sID> register <cID>
```

- دانشجوی با شماره‌ی sID در درس cID ثبت‌نام می‌کند.

```
<sID> register <cID1> <cID2> ...
```

- دانشجوی با شماره‌ی sID در درس‌های $cID1$ و $cID2$ و ... ثبت‌نام می‌کند.

```
<ltID> capacitate <cID> <n>
```

- استاد با شناسه‌ی $ltID$ ظرفیت درس cID را به اندازه‌ی n افزایش می‌دهد.

```
<ltID> mark <cID> <sID1> <mark1> <sID2> <mark2> ...
```

- استاد با شناسه‌ی $ltID$ که درس cID را ارائه می‌دهد، برای دانشجوی $sID1$ نمره‌ی $mark1$، برای دانشجوی $sID2$ نمره‌ی $mark2$ و.... را ثبت می‌کند. تضمین می‌شود نمره‌ی تمام دانشجویان درس cID وارد می‌شود.

```
<ltID> mark <cID> <mark> -all
```

- استاد با شناسه‌ی $ltID$ برای دانشجویان درس cID نمره‌ی $mark$ را رد می‌کند.

```
end semester
```

- پایان ترم جاری

```
start semester
```

- شروع ترم جاری

```
end registration
```

- پایان زمان ثبت‌نام

```
addCourse <cID> <unit>
```

- درس با شناسه‌ی cID که $unit$ واحد دارد به سامانه اضافه می‌شود.

دستورات $show$:

```
showCourse <cID> <students|lecturer|capacity|average>
```

برای درس cID ویژگی خواسته شده نمایش داده می‌شود. (اگر $students$ بود لیست تمامی دانشجویان ثبت‌نامی به ترتیب شماره‌ی دانشجویی، اگر $lecturer$ بود، شناسه‌ی استاد ارائه‌دهنده‌ی درس، اگر $capacity$ بود، ظرفیت درس و اگر $average$ بود، معدل درس را چاپ کند.). در صورت معتبر نبودن شماره‌ی درس یا حذف شدن آن (به حد نصاب نرسیدن)، پیغام $shoma\ daneshjoo\ nistid$ را چاپ کنید.

```
showRanks <cID>
```

- شناسه‌ی سه دانشجویی که بالاترین نمرات درس را کسب کرده‌اند چاپ می‌کند. (در صورت برابر بودن نمره‌ی چند دانشجو، سه نفر اول به ترتیب شماره‌ی دانشجویی چاپ شوند.)

```
showAverage <sID>
```

- معدل یک دانشجو را چاپ می‌کند. در صورت وجود نداشتن دانشجوی با شناسه‌ی sID، پیام $shoma\ daneshjoo\ nistid$ را چاپ کنید.

```
showRanks -all
```

- نمایش دانشجویان با رتبه‌ی ۱ تا ۳ از نظر معدل. (در صورت برابر بودن، به ترتیب شماره‌ی دانشجویی چاپ شوند.)

نکات دیگر

- هر درس به طور پیش‌فرض دارای ۱۵ نفر ظرفیت می‌باشد و باید دارای حداقل ۳ نفر برای تشکیل باشد.
- هر دانشجو باید حداقل ۱۲ واحد در یک ترم داشته باشد.
- هر درس تنها توسط یک استاد ارائه می‌شود.
- هر دانشجو تنها ۳ واحد را می‌تواند W کند به این صورت که تا زمانی که واحدهایی که درخواست w آن‌ها را داده از ۳ واحد بیشتر نیستند، آن‌ها حذف می‌شوند و سپس دیگر حذف نمی‌شوند. مثلا اگر دستور حذف یک درس ۴ واحدی را وارد کند، نه‌تنها درس حذف نمی‌شود بلکه هیچ واحد دیگری نیز نمی‌تواند حذف کند.
- شناسه‌ی اساتید و دانشجویان ۵ رقمی هستند.
- با شروع ترم، ثبت‌نام نیز شروع می‌شود.
- پس از پایان زمان ثبت‌نام، ابتدا تمام دروسی که به حد نصاب نرسیده‌اند حذف شده و دانشجویان دیگر آن درس‌ها را نخواهند داشت. سپس تعداد واحد تمام دانشجویان چک می‌شود. پس از این مرحله دیگر درس‌ها چک نشده و ترم رسما آغاز می‌گردد.
- در صورتی که یک دانشجو از کف مجاز واحد کمتر داشته باشد، ترم وی حذف می‌شود.
- دستورات اضافه شدن دانشجو و استاد قبل شروع ترم وارد می‌شوند.
-پس از اتمام ترم دستورات $show$ وارد می‌شوند و برنامه با دستور $endShow$ پایان می‌پذیرد.
- در صورت وارد شدن دستورات بی‌ربط چیزی چاپ نمی‌شود.
- تمامی محاسبات اعشاری با دقت یک رقم اعشار محاسبه شود.

مثال

ورودی	خروجی
addCourse 40101 3 addCourse 40102 4 addCourse 40103 4 addCourse 40104 3 addCourse 25001 4 addCourse 25002 3 addCourse 25003 4 addCourse 65001 3 addCourse 65002 3 addCourse 65003 4 addStudent 95101 addStudent 95102 addStudent 95103 addStudent 95103 addStudent 95104 addStudent 94101 addStudent 94102 addStudent 94103 addLecturer 10001 40101 40103 addLecturer 10002 40102 40104 addLecturer 10003 25001 25002 addLecturer 10004 65001 65002 65003 start semester 95101 register 40101 40102 40103 25001 25002 95102 register 40101 40102 40103 25001 25002 95103 register 40101 40102 40103 25001 25002 95104 register 40101 40102 40103 25002 65002 94101 register 40102 40103 25001 25002 65001 94102 register 40102 40103 25001 25002 65001 94103 register 40102 40103 25001 65001 end registration 10001 mark 40101 95101 12 95102 18 95103 18.5 95104 9.9 10002 mark 40102 95101 19.8 95102 20 95103 16.5 95104 19 94101 17 94102 10 94103 15.9 10001 mark 40103 95101 17.4 95102 19 95103 15.7 95104 11.2 94101 19 94102 17.8 94103 20 10003 mark 25001 95101 9 95102 10 95103 10 94101 10 94102 19 94103 12 10003 mark 25002 95101 20 95102 14 95103 18 95104 9.5 94101 20 94102 18 94103 19.6 10004 mark 65001 94101 18 94102 10 94103 15 W 40101 95101 end semester showAverage 95103 showCourse 25001 average showRanks 65001 endShow	15.5 11.7 94101 94103 94102

۲ دوز کذایی!! (۳۵ امتیاز)

در این سوال شما باید بازی دوز یا همان XO را پیاده‌سازی کنید.

ویژگی‌های بازی:

- قابلیت بازی در صفحه‌ی $n \times m$ که $3 \leq n, m$ که اگر n یا m برابر ۳ بودند برنده با چیدن ۳ X یا O به صورت افقی یا عمودی یا قطری برنده می‌شود و اگر هر دو بزرگتر از ۳ بودند با ۴ X یا O این اتفاق می‌افتد. (به صورت پیش‌فرض جدول بازی ۳ در ۳ است.)
- تعریف کاربر به این صورت که هنگام شروع بازی نام دو بازیکن گرفته می‌شود و اگر حداقل یکی از این دو بازیکن وجود نداشتند، بازیکن(ان) جدیدی به آن نام(ها) ایجاد شده و وارد بازی می‌شوند. نتیجه‌ی برد یا باخت یا تساوی بازی‌ها برای هر بازیکن ذخیره می‌شود.
- جدول امتیازات: در جدول امتیازات تعداد بردها، باخت‌ها و تساوی‌های تمامی بازیکنان نشان داده می‌شود که بازیکنان به ترتیب تعداد برد نمایش داده می‌شوند (اگر دو کاربر تعداد برد یکسان داشتند هر کدام تعداد باخت کمتری داشته باشند اول می‌آید. در صورت تساوی تعداد بردها و باخت‌ها هر کدام تعداد تساوی کمتری داشتند اول می‌آید و اگر تعداد بردها و باخت‌ها و تساوی‌های دو بازیکن یکی بود بر اساس نام بازیکن سورت می‌شوند.) هم‌چنین در صورتی که هیچ بازیکنی در بازی وجود نداشت چیزی در جدول امتیازات چاپ نمی‌شود.
- قابلیت $undo$: هر کاربر در طول هر بازی حداکثر می‌تواند یک بار از $undo$ استفاده کند که با این کار مهره‌ای که در حرکت قبلی گذاشته را می‌تواند بردارد. توجه کنید که دستور $undo$ برای یک بازیکن در نوبت بازیکن حریف وارد می‌شود و با این کار نوبت به بازیکنی که از $undo$ استفاده کرده برمی‌گردد.
- قابلیت توقف و ادامه‌ی بازی: بازیکنان می‌توانند بازی را با دستور $pause$ متوقف کرده و با این کار بازی به منوی اصلی برمی‌گردد. برای ادامه‌ی بازی در منوی اصلی از قسمت بازی‌های ناتمام بازی را انتخاب کرده و آن را ادامه می‌دهیم.

دستورات

در منوی اصلی بازی

new game $<$ Player1 $>$ $<$ Player2 $>$

این دستور یک بازی جدید با کاربران $Player1$ و $Player2$ ایجاد می‌کند که اگر یکی از این بازیکنان یا هر دو وجود نداشتند، کاربر جدید ایجاد شده و وارد بازی می‌شوند. به $Player1$ مهره‌ی X و به $Player2$ مهره‌ی O نسبت داده می‌شود و بازیکن $Player1$ بازی را شروع می‌کند.
- مثال:

```
new game ali ahmad
```

در صورتی که نام یک یا دو کاربر جلوی دستور $new\ game$ وارد نشود پیغام خطای $Invalid\ players$ چاپ شود. تضمین می‌شود که نام هر بازیکن یک تکه است و دارای فاصله نیست.

resume

با این دستور وارد منوی $resume$ می‌شویم.

set table n $*$ m

با این دستور، جدول بازی تبدیل به جدول $n \times m$ می‌شود (n ردیف و m ستون) و تا زمانی که دوباره این مقدار تغییر نکند، تمام بازی‌های جدید در جدول $n \times m$ انجام می‌گیرند. (تضمین می‌شود مقادیر n و m حداقل ۳ هستند.)
- مثال:

```
set table 5*4
```

در صورتی که جلوی دستور $set\ table$ هیچ عددی وارد نشود، جدول به حالت پیش‌فرض ۳ در ۳ تبدیل می‌شود. این دستور برای بازی‌های جدید اعمال می‌شود و برای بازی‌هایی که قبل از این دستور ذخیره شده‌اند باید جدول بازی با همان تنظیمات قبلی ادامه پیدا کند.

scoreboard

با زدن این دستور وارد جدول امتیازات می‌شویم.

quit

با وارد کردن این دستور از بازی خارج می‌شویم و برنامه بسته می‌شود.

در جدول امتیازات

با ورود به جدول امتیازات، به ترتیب ذکر شده در ویژگی‌های بازی، جدول امتیازات نمایش داده می‌شود که در آن عدد اول تعداد بردها، عدد دوم تعداد باخت‌ها و عدد سوم تعداد تساوی‌های بازیکن است. (به مثال توجه کنید.)
مثال:

```
ali 4 3 3
ahmad 3 4 3
abbas 3 5 2
mohammadali 2 6 2
mohammadhossein 2 6 2
```

back

با این دستور از محیط جدول امتیازات خارج می‌شویم و به منوی اصلی برمی‌گردیم.

quit

با وارد کردن این دستور از بازی خارج می‌شویم و برنامه بسته می‌شود.

در منوی $resume$

در این منو لیستی از بازی‌های ناقص نمایش داده می‌شود (بر حسب زمان انجام بازی). با وارد کردن شماره‌ی بازی وارد محیط بازی شده و ادامه‌ی بازی را انجام می‌دهیم. در صورت ورود عدد اشتباه پیغام $Invalid\ number$ نمایش داده می‌شود.

$(no).\ Player1\ Player2$

مثال:

```
1. ali ahmad
2. ahmad abbas
3. ali abbas
```

توضیح مثال: بازی بین علی و عباس اولین بازی $pause$ شده است و جدیدترین بازی $pause$ شده بازی بین علی و احمد است. با وارد کردن عدد 1 ادامه‌ی بازی بین ali و $ahmad$ انجام می‌گیرد.

back

با ورود این دستور به منوی اصلی برمی‌گردیم.

در محیط بازی

با ورود به محیط بازی، جدول بازی به همراه نوبت کاربر نمایش داده می‌شود (خانه‌های خالی با کاراکتر '_' نمایش داده می‌شوند). مثال برای جدول ۳ در ۳:

```
_|_|_
_|_|_
_|_|_
ali
```

put $(\mathbf{x},\ \mathbf{y})$

با واردکردن این دستور بسته به اینکه نوبت کدام بازیکن باشد، مهره‌ی X یا O در ردیف x و ستون y گذاشته می‌شود. پس از این دستور جدول بازی نمایش داده می‌شود.
مثال:

```
X|O|_
X|_|_
_|_|_
```

اگر حرکتی بازی را اتمام بخشید، به جای جدول بازی پیغام مناسب نمایش داده می‌شود. (در صورت تساوی پیغام $Draw$ و در صورت بردن بازیکن با نام x پیغام $Player\ x\ won$ نمایش داده می‌شود.) پس از نمایش پیغام مناسب، به منوی اصلی برمی‌گردیم.
مثال مختصات:

(1, 1)	(1, 2)	(1, 3)	(1, 4)
(2, 1)	(2, 2)	(2, 3)	(2, 4)
(3, 1)	(3, 2)	(3, 3)	(3, 4)

در صورتی که مختصات وارد شده خارج از جدول بود یا خانه‌ی انتخاب شده پر بود، پیغام $Invalid\ coordination$ چاپ شود.

undo

با وارد شدن این دستور آخرین مهره‌ای که گذاشته شده برداشته می‌شود و نوبت بازی به بازیکن نوبت قبل برمی‌گردد. (توجه شود که زمانی که نوبت بازی بازیکن اول است، با زدن دستور $undo$ مهره‌ی قبلی برداشته شده و نوبت به بازیکن دوم می‌رسد و بازیکن دوم دیگر نمی‌تواند از دستور $undo$ استفاده کند.) پس از اجرای این دستور جدول بازی پس از برداشتن مهره‌ی آخر نمایش داده می‌شود.
این دستور از زمانی قابل استفاده است که از هر مهره حداقل یک عدد در جدول موجود باشد و در غیر اینصورت، با ورود این دستور پیغام خطای $Invalid\ undo$ چاپ می‌شود و جدول بازی دوباره چاپ می‌شود. در صورتیکه بازیکن از فرصت $undo$ی خود استفاده کرده بود با وارد شدن مجدد دستور $undo$ همین پیغام خطا چاپ شود.

pause

با وارد شدن این دستور، بازی ذخیره شده و به منوی اصلی برمی‌گردیم.

stop

با وارد شدن این دستور، بازی بدون نتیجه مانده و به منوی اصلی بازمی‌گردیم (به تعداد بردها و باخت‌ها و تساوی‌های بازیکنان چیزی اضافه نمی‌شود.)

نکات

- هنگام انجام بازی، اگر دستوری با پیغام $Invalid$ مواجه شد، پس از نمایش پیغام، مجددا جدول بازی و نوبت کاربر در خروجی نمایش داده می‌شود. (به ورودی و خروجی نمونه توجه کنید.)
- هر دستوری غیر از موارد ذکر شده در بالا، منجر به چاپ خطای $Invalid\ command$ می‌شود. (این نکته لازم به ذکر است که دستورات هر بخش فقط برای همان بخش هستند و مثلا وارد کردن دستور $scoreboard$ زمانی که بازیکنان مشغول بازی کردن هستند منجر به تولید خطای $Invalid\ command$ می‌شود.
- هم‌چنین برای دستورات ناقص اگر در توضیحات بالا صریحا خطایی ذکر شده همان خطا چاپ می‌شود در غیر اینصورت پیغام $Invalid\ command$ چاپ شود.

ورودی مثال ۱

```
new game ali
```

خروجی مثال ۱

```
Invalid Players
```

ورودی مثال ۲

```
new game ali ahmad abbas
```

خروجی مثال ۲

```
Invalid command
```

مثال

ورودی نمونه

```
new game hamid hamed
put(2,2)
pause
set table 4*4
new game hadi mahdi
put(3,1)
stop
scoreboard
back
resume
back
new game hamid hadi
put(1,1)
put(1,2)
put(2,1)
put(2,2)
put(3,1)
put(3,2)
pause
resume
2
put(4,1)
pause
resume
1
put(4,1)
scoreboard
back
quit
```

خروجی نمونه

```
_|_|_
_|_|_
_|_|_
hamid
_|_|_
_|X|_
_|_|_
hamed
_|_|_|_
_|_|_|_
_|_|_|_
_|_|_|_
hadi
_|_|_|_
_|_|_|_
X|_|_|_
_|_|_|_
mahdi
hadi 0 0 0
hamed 0 0 0
hamid 0 0 0
mahdi 0 0 0
1. hamid hamed
_|_|_|_
_|_|_|_
_|_|_|_
_|_|_|_
hamid
X|_|_|_
_|_|_|_
_|_|_|_
_|_|_|_
hadi
X|O|_|_
_|_|_|_
_|_|_|_
_|_|_|_
hamid
X|O|_|_
X|_|_|_
_|_|_|_
_|_|_|_
hadi
X|O|_|_
X|O|_|_
_|_|_|_
_|_|_|_
hamid
X|O|_|_
X|O|_|_
X|_|_|_
_|_|_|_
hadi
X|O|_|_
X|O|_|_
X|O|_|_
_|_|_|_
hamid
1. hamid hadi
2. hamid hamed
_|_|_
```

```
_|X|_
_|_|_
hamed
Invalid coordination
_|_|_
_|X|_
_|_|_
hamed
1. hamid hamed
2. hamid hadi
X|O|_|_
X|O|_|_
X|O|_|_
_|_|_|_
hamid
Player hamid won
hamid 1 0 0
hamed 0 0 0
mahdi 0 0 0
hadi 0 1 0
```

فیفا یا پس؟ مسئله این است!!!

در این سوال، باید یک لیگ فوتبال (مانند بازی فیفا) را پیاده سازی کنید. در این لیگ، تعدادی تیم وجود دارد که هر تیم از چند بازیکن تشکیل شده است.

لیگ

همان‌طور که اشاره شد، در این لیگ، تعدادی تیم وجود دارد. و همچنین بازیکن‌هایی که تیم ندارند در لیگ قرار می‌گیرند. در واقع، تمام کارهای اصلی (از جمله خرید و فروش بازیکن، انجام بازی‌های دوستانه، و سپری کردن یک فصل از لیگ) در سازمان لیگ انجام می‌شود.

تیم

هر تیم، یک نام، مقداری پول به عنوان بودجه، یک ترکیب اصلی و تعدادی بازیکن ذخیره دارد.

ترکیب اصلی (Squad)

در ترکیب اصلی، ۱۱ بازیکن که قرار است در بازی‌ها حاضر شوند، قرار می‌گیرند. برای راحتیِ کار، تمام ترکیب ها از آرایش ۴-۳-۳ برخوردار هستند؛ که یعنی ۱ دروازه‌بان، ۴ مدافع، ۳ هافبک و ۳ مهاجم.

بازیکن

هر بازیکن، نام، نام خانوادگی، سن، و یک قرارداد دارد.

۴ پست مختلف برای بازیکن‌ها وجود دارد؛ دروازه‌بان، مدافع، هافبک، و مهاجم. هر بازیکن به نسبت پست خودش، معیارهایی برای سنجش توانایی‌هایش دارد.

دروازه‌بان

معیارهای هر دروازه‌بان عبارت اند از:

- قدرت شوت گیری (Shot Saving)
- واکنش نشان دادن (Reactions)
- پنالتی‌گیری (Penalty Saving)

مدافع

معیارهای هر مدافع عبارت اند از:

- قدرت بدنی (Strength)
- خشونت (Aggression)
- سر زنی (Heading)

هافبک

معیارهای هر هافبک عبارت اند از:

- پاس‌کاری (Passing)
- شوت زنی (Shooting)
- سانتر کردن (Crossing)

مهاجم

معیارهای هر مهاجم عبارت اند از:

- سرزنی (Heading)
- تمام کنندگی (Finishing)
- پنالتی زنی (Penalties)

هر بازیکن در زمان ایجاد شدن (پیوستن به دنیای فوتبال)، یک بازیکن آزاد محسوب می‌شود. زمانی که بازیکن به تیمی بپیوندد، قراردادی امضا می‌کند که در آن قرارداد، تعداد سال‌هایی که بازیکن با آن تیم قرارداد دارد ذکر شده است. همچنین یک نوع قرارداد قرضی نیز داریم که یک بازیکن در صورتی که در یک تیم مشغول به فعالیت باشد، می تواند برای مدتی در تیمی دیگر بازی کند و پس از اتمام قرارداد قرضی، به تیم اولیه خود بازگردد. مدت قرارداد قرضی نباید از مدت قرارداد اصلی بیشتر باشد.

در هر فصل از لیگ، هر تیم با تمام تیم‌های دیگر دو بار بازی می‌کند. یک بار در خانه‌ی خودش و بار دیگر در خانه‌ی حریف. در صورتی که تیمی در یک بازی مهمان باشد، از معیار های هر بازیکنِ آن ۵ تا کم می شود.

هر تیم، مقداری بودجه دارد که از آن برای خرید بازیکن استفاده می‌کند. هزینه‌ی خرید بازیکن‌ها از بودجه‌ی تیم مقصد پرداخت شده و به بودجه‌ی تیم مبدا واریز می‌شود و بازیکن خریداری شده به لیست بازیکنان ذخیره‌ی تیم جدید منتقل می‌شود. بودجه‌ی تیم‌ها و هزینه‌های مشخص شده همگی بر حسب میلیون دلار هستند.

نتیجه هر بازی به این صورت محاسبه می‌شود:

- در صورتی که میانگین قدرت پاس‌کاری هافبک‌ها از ۸۵ بیشتر باشد، موقعیت های بهتری نصیب مهاجم ها می‌شود و به قدرت تمام‌کنندگی هر مهاجم ۵ تا اضافه می‌شود. همچنین اگر هافبکی در تیم باشد که قدرت پاس‌کاری‌اش از ۹۰ بیشتر باشد، به قدرت تمام کنندگی هر مهاجم ۵ تا اضافه می‌شود.
- در صورتی که میانگین قدرت بدنی مدافع‌های حریف از ۸۵ بیشتر باشد، آن‌ها در مصاف‌هایی که در مقابل مهاجمان دارند پیروز می‌شوند و از قدرت تمام‌کنندگی هر مهاجم ۳ تا کم می‌شود. همچنین اگر مدافعی در تیم باشد که قدرت بدنی‌اش از ۹۰ بیشتر باشد، از قدرت تمام‌کنندگی هر مهاجم ۳ تا کم می‌شود.
- به ازای هر مهاجم، اگر قدرت تمام‌کنندگی او از قدرت واکنش نشان دادن دروازه‌بان حریف بیشتر باشد، آن مهاجم یک گل به ثمر می‌رساند. و اگر میانگین قدرت تمام‌کنندگی مهاجم‌ها از قدرت واکنش نشان دادن دروازه بان حریف حداقل ۵ تا بیشتر باشد، تیم حمله کننده ۲ گل به ثمر می‌رساند.
- به ازای هر هافبک، اگر قدرت شوت‌زنی او از قدرت شوت‌گیری دروازه‌بان حریف بیشتر باشد، او یک گل به ثمر می‌رساند .
- به ازای هر مدافع یا مهاجم، اگر قدرت سرزنی او ضرب در میانگین قدرت سانتر هافبک ها، از مجذور قدرت واکنش نشان دادن دروازه‌بان حریف بیشتر باشد، آن بازیکن یک گل به ثمر می‌رساند.
- اگر میانگین خشونت مدافعان حریف از ۸۵ بیشتر باشد، تیم حمله‌کننده صاحب یک ضربه پنالتی می‌شود و مهاجمی که بیشترین قدرت پنالتی‌زنی را دارد پشت توپ می‌ایستد. در صورتی که قدرت پنالتی‌زنی او از قدرت پنالتی‌گیری دروازه‌بان حریف بیشتر باشد، پنالتی تبدیل به گل می‌شود، و در غیر این‌صورت دروازه‌بان پنالتی را می‌گیرد.
- پس از محاسبه‌ی تعداد گل‌های هر تیم، برنده و بازنده‌ی آن بازی (یا تساوی تیم‌ها) مشخص می‌شود. هر برد ۳ امتیاز و تساوی ۱ امتیاز دارد. باخت هم امتیازی ندارد.
- در انتهای فصل، به تیم‌های اول، دوم و سوم به ترتیب ۱۰۰، ۵۰ و ۲۰ میلیون دلار پاداش داده می‌شود. بازیکنانی که قراردادشان تمام شده به لیست بازیکنان آزاد منتقل می‌شوند و بازیکنانی که با قرارداد قرضی در تیم دیگری بازی می‌کردند به تیم اصلی بازمی‌گردند. همچنین سن تمام بازیکنان یک سال افزایش می‌یابد. بازیکنانی که ۴۰ سال یا بیشتر سن دارند، بازنشسته شده و از فوتبال خداحافظی می‌کنند.

دستورات ورودی

create team teamName

یک تیم با نام مشخص شده می‌سازد. بودجه‌ی اولیه‌ی تیم ۱۰۰ میلیون دلار است.
در صورتی که تیمی با این نام وجود داشته باشد، $a\ team\ exists\ with\ this\ name$ چاپ می‌شود.
در صورتی که تیم با موفقیت ساخته شود $team\ created$ چاپ می‌شود.

---

delete team teamName

تیم مورد نظر را منحل می‌کند. در صورتی که تیمی با این نام وجود نداشته‌باشد، $invalid\ team$ چاپ می‌شود.
پس از حذف تیم مورد نظر، بازیکنان آن به بازیکنان آزاد منتقل می‌شوند و قراردادهای آن‌ها نیز فسخ می‌شود.

---

create player firstName lastName age position

دقت کنید که position باید با یکی از عبارات GK (دروازه بان)، DF (مدافع)، MF (هافبک)، و یا ST (مهاجم) برابر باشد.
اگر بازیکنی با این نام وجود داشته باشد $a\ player\ exists\ with\ this\ name$ چاپ می‌شود.
اگر ورودی مشکلی نداشت، به نسبت پست بازیکن، معیارها را دریافت می‌کند. به این صورت که به ترتیب آورده شده در بالا، معیارها چاپ می‌شود و مقادیر مورد نظر از ورودی دریافت می شود. مثلا برای یک مدافع، عبارات زیر در خروجی چاپ شده و پس از چاپ شدن هر خط، مقدار مربوطه برای هر معیار دریافت می‌شود:

Strength:
Aggression:
Heading:

زمانی که بازیکن که ساخته شد $player\ created$ چاپ شده و بازیکن به لیست بازیکن‌های آزاد منتقل می‌شود.

---

print free agents

لیست بازیکن‌های آزاد را ابتدا به بر اساس پست (به ترتیب دروازه‌بان، مدافع، هافبک، مهاجم)، و سپس نام خانوادگی و نام مرتب کرده و چاپ می‌کند.
مثال:

```
1.  Mohammad Mohammadi GK
2.  Ali Alavi DF
3.  Reza Rezaei DF
...
```

---

move free agent firstName lastName to teamName with #years years contract

در صورتی که تیم یافت نشود $invalid\ team$ چاپ می‌شود.
در صورتی که بازیکن آزاد مورد نظر یافت نشود $invalid\ free\ agent$ چاپ می‌شود.
در صورت قابل قبول بودن ورودی‌ها، بازیکن آزاد مورد نظر با قرارداد years ساله به تیم مشخص شده منتقل می‌شود و پیام $free\ agent\ moved$ چاپ می‌شود.

---

print stats of team teamName

مشخصات یک تیم از جمله نام تیم، بودجه‌ی آن، و بازیکنان اصلی و ذخیره آن تیم را چاپ می‌کند. در صورتی که تیم یافت نشود $invalid\ team$ چاپ می‌شود.
ترتیب بازیکن‌ها باید بر اساس پست (دروازه‌بان، مدافع، هافبک، مهاجم)، و سپس نام خانوادگی و نام مرتب شود.
مثال:

```
Team: Chelsea
Budget: 100
Squad players:
1.  Kepa Arrizabalaga GK Age: 24 Contract: 6 years
2.  Marcos Alonso DF Age: 28 Contract: 4 years
3.  Cesar Azpilicueta DF Age: 29 Contract: 3 years
Reserve players:
1.  Willy Caballero GK Age: 37 Contract: 1 years
2.  Gary Cahill DF Age: 33 Contract: 1 years
```

---

print stats of player firstName lastName

مشخصات یک بازیکن را چاپ می‌کند. در صورتی که بازیکن نامعتبر باشد $invalid\ player$ چاپ می‌شود.
مثال:

```
Eden Hazard ST
Age: 28
Heading: 61
Finishing: 84
Penalties: 86
```

---

renew contract of firstName lastName for #years years

قرارداد بازیکن مورد نظر را به مدت مشخص شده تمدید می‌کند. در صورتی که بازیکن مورد نظر، قراردادی نداشته باشد، پیغام $invalid\ renewal\ command$ چاپ می‌شود. در صورتی که بازیکن یک قرارداد قرضی داشته باشد، قراردادش با تیم اصلی تمدید می‌شود. در صورت موفقیت آمیز بودن تمدید، پیام زیر چاپ می شود:

$$contract\ renewed.\ new\ contract\ is\ valid\ for\ \#years\ years$$

در صورتی که بازیکن یافت نشود، پیغام $invalid\ player$ چاپ می‌شود.

---

terminate contract of firstName lastName

قرارداد بازیکن مورد نظر را فسخ می‌کند. در صورتی که بازیکن مورد نظر، قراردادی نداشته باشد، پیغام $invalid\ termination\ command$ چاپ می‌شود. در صورتی که بازیکن یک قرارداد قرضی داشته باشد، قرارداد قرضی‌اش فسخ می‌شود. در صورتی که بازیکن یافت نشود، پیغام $invalid\ player$ چاپ می‌شود.

---

sell player firstName lastName from sourceTeam to destinationTeam for #price$ with #years years contract

بازیکن مورد نظر را از تیم مبدا به تیم مقصد با مبلغ و طول قرارداد مشخص شده منتقل می‌کند. در صورتی که تیم مبدا نامعتبر باشد، $invalid\ source\ team$ چاپ می‌شود. در غیر این صورت، اگر بازیکن مورد نظر در تیم مبدا وجود نداشته باشد، $invalid\ player$ چاپ می‌شود. در صورتی که تیم مبدا و بازیکن معتبر باشند ولی تیم مقصد نامعتبر باشد $invalid\ destination\ team$ چاپ می‌شود.
در صورتی که تیم مقصد بودجه‌ی لازم برای انتقال را نداشته باشد، $insufficient\ budget$ چاپ می‌شود.

---

loan player firstName lastName from sourceTeam to destinationTeam with #years years contract

بازیکن مورد نظر از تیم مبدا به تیم مقصد و با مدت قرارداد مشخص شده قرض داده می‌شود. پیام‌های چاپ شده در صورت نامعتبر بودن ورودی‌ها مانند $sell$ هستند.
اگر طول قرارداد قرضی از طول قرارداد اصلی بیشتر بود پیام $invalid\ loan\ contract$ چاپ شود.

---

put player firstName lastName from teamName in main squad

بازیکن مورد نظر از تیم مشخص شده را در ترکیب اصلی قرار می‌دهد. در صورتی که تیم نامعتبر باشد $invalid\ team$ چاپ می‌شود. در صورتی که بازیکن در ترکیب اصلی تیم باشد، $player\ is\ currently\ in\ the\ main\ squad$ چاپ می‌شود. در صورتی که بازیکن نه در ترکیب اصلی و نه در ذخیره‌ها باشد، $invalid\ player$ چاپ می‌شود. در غیر این صورت، بازیکن به ترکیب اصلی (و به پست خودش) منتقل می‌شود. نحوه‌ی قرار گرفتن بازیکن ها در ترکیب تیم ها به صورت $First\ in - First\ out$ است، یعنی در صورتی که بازیکنان مربوط به یک پست تکمیل باشند، اولین بازیکنی که در آن پست به ترکیب اضافه شده، از ترکیب خارج می‌شود و بازیکن مورد نظر به ترکیب اضافه می‌شود.

---

friendly match between team1 and team2

یک بازی دوستانه میان تیم اول و تیم دوم برگزار می‌کند. در این حالت از بازی، اثرات مربوط به بازی در خانه حریف اعمال نمی‌شود. در صورتی که حداقل یکی از تیم ها نامعتبر باشد، $invalid\ team$ چاپ می‌شود. در صورتی که هر تیم، ترکیب ناقصی داشته باشد، عبارت زیر (به نسبت هر تیمی که ترکیبش ناقص است) چاپ می‌شود:

$$team1/team2\ squad\ isn't\ complete$$

در پایان بازی، نتیجه بازی به صورت زیر چاپ می‌شود:

$$team1\ \#team1Goals\ -\ \#team2Goals\ team2$$

---

next season

با این دستور، بازی های مربوط به فصل بعد انجام می‌شود و اعمال انتهای فصل (توضیح داده شده در بالا) انجام می‌شوند. در صورتی که ترکیب اصلی تیمی ناقص بود، به ترتیب الفبایی برای نام تیم‌ها، به ازای آن‌ها در خروجی عبارت $teamName\ squad$ $isn't\ complete$ چاپ می‌شود. در انتها جدول لیگ چاپ می شود. ترتیب قرار گرفتن تیم‌ها در جدول به این صورت است که اولویت ابتدا با امتیاز است. سپس تفاضل گل و گل زده شده بررسی می شود و در صورتی که تمام موارد ذکر شده برابر بودند، تیم ها بر اساس ترتیب نام‌هایشان در جدول قرار می‌گیرند!
یک نمونه برای جدول:

```
1.  Chelsea 95 points W: 29 D: 8 L: 1 GF: 72 GA: 15 GD: +57
2. …
```

اعداد ذکر شده در جدول، به ترتیب امتیاز تیم، تعداد بردها، تعداد تساوی‌ها، تعداد باخت‌ها، تعداد گل‌های زده، تعداد گل‌های خورده و تفاضل گل هستند.

---

end

با خواندن این دستور، بازی پایان می‌یابد.

نکته: هر دستور غیر از موارد ذکر شده منجر به تولید $invalid\ command$ می‌شود.