

گلستان

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

مینی پروژه شماره 2 : گلستان

سامانه گلستان، یک سامانه بسیار قوی و کارآمد است که برای وضعیت تحصیلی دانشجویها و برگزاری و ثبت نام کلاسهای دانشجویها طراحی شده است.

هر دانشجو و هر استاد به محض ورود به دانشگاه در سامانه گلستان ثبت نام می‌کنند. در این سامانه مشخصات هر کدام وارد می‌شود. همچنین تعدادی کلاس در سامانه گلستان تعریف می‌شود. هر دانشجو می‌تواند در تعدادی از این کلاسها ثبت نام کند.

اساتید نمرات را در سامانه گلستان وارد می‌کنند و دانشجویان هم در این سامانه می‌توانند نمرات خود را در این سامانه ببینند.

▼ دانشجو

• اسم name

◦ یک رشته به طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

• کد ملی identical_num

◦ یک رشته 10 رقمی از ارقام 0 تا 9 است.

• سال ورودی entering_year

◦ یک عدد صحیح بین 1300 تا 1500 است. (شامل دو سر بازه)

• رشته تحصیلی field

◦ یک رشته به طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

▼ استاد

• اسم `name`

◦ یک رشته طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

• کد ملی `identical_num`

◦ یک رشته 10 رقمی از ارقام 0 تا 9 است.

• رشته تحصیلی `field`

◦ یک رشته به طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

▼ کلاس

• اسم `name`

◦ یک رشته به طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

• کد کلاس `class_id`

◦ یک رشته 10 رقمی از ارقام 0 تا 9 است.

• رشته `field`

◦ یک رشته به طول حداقل 1 و حداکثر 20 از حروف کوچک انگلیسی است.

• استاد `professor`

◦ یک استاد برای تدریس این درس است.

• دانشجویان `students`

◦ یک لیست از دانشجویانی که این درس را می‌آموزند است.

از شما می‌خواهیم تا به صورت مرحله به مرحله این سامانه را طراحی کنید. در واقع در پیاده سازی هر فاز از دستورات فاز قبلی هم استفاده خواهد شد. در این مینی پروژه به پیاده سازی فاز اول و دوم آن خواهیم پرداخت .

▼ فاز اول

▼ درخواست `register_student`

```
register_student <name> <identical_num> <entering_year> <field>
```

در این درخواست یک دانشجو نام <name>، کد ملی <identical_num>، سال ورودی <entering_year>، رشته تحصیلی <field> خود را وارد می کند و اقدام به ثبت نام در سامانه گلستان می کند.

در صورتی که یک دانشجو یا استاد دیگر قبلاً با کد ملی (identical_num) مشابه در سیستم ثبت نام کرده است پیام this identical number previously registered را چاپ کنید.

بعد از ثبت نام با موفقیت پیام welcome to golestan را چاپ کنید.

▼ درخواست register_professor

```
register_professor <name> <identical_num> <field>
```

در این درخواست یک استاد نام <name>، کد ملی <identical_num>، رشته تدریس <field> خود را وارد می کند و اقدام به ثبت نام می کند.

در صورتی که یک دانشجو یا استاد دیگر قبلاً با کد ملی (identical_num) مشابه در سیستم ثبت نام کرده است پیام this identical number previously registered را چاپ کنید.

بعد از ثبت نام با موفقیت پیام welcome to golestan را چاپ کنید.

▼ درخواست make_class

```
make_class <name> <class_id> <field>
```

در این درخواست یک کلاس با نام <name> و کد درس <class_id> برای رشته تحصیلی <field> ایجاد می شود.

اگر این کد کلاس دیگری قبلاً استفاده شده است، پیام this class id previously used را چاپ

کنید.

در غیر این صورت پیام `class added successfully` را چاپ کنید.

▼ فاز دوم

▼ درخواست `add_student`

`add_student <identical_num> <class_id>`

در این درخواست یک دانشجو با کد ملی `<identical_num>` درخواست اضافه شدن به کلاس با کد `<class_id>` را دارد.

- در صورتی که چنین دانشجویی وجود ندارد پیام `invalid student` را چاپ کنید.
- در صورتی که چنین کلاسی وجود ندارد پیام `invalid class` را چاپ کنید.
- در صورتی که رشته تحصیلی این دانشجو با `<field>` این کلاس یکسان نیست پیام `student field is not match` را چاپ کنید.
- در صورتی که این دانشجو قبلاً در این کلاس ثبت نام کرده است پیام `student is already registered` را چاپ کنید.
- در صورتی که هیچ کدام از حالت‌های فوق پیش نیامد پیام `student added successfully` to the class را چاپ کنید.

توجه کنید اگر چند خطا همزمان اتفاق افتاد، پیام‌ها از بالا به پایین الویت دارند.

▼ درخواست `add_professor`

`add_professor <identical_num> <class_id>`

در این درخواست یک استاد با کد ملی `<identical_num>` می‌خواهد تدریس کلاس با کد `<class_id>` را به عهده بگیرد.

- در صورتی که چنین استادی وجود ندارد پیام `invalid professor` را چاپ کنید.
 - در صورتی که چنین کلاسی وجود ندارد پیام `invalid class` را چاپ کنید.
 - در صورتی که رشته تدریس این استاد با `<field>` این کلاس یکسان نیست پیام `professor field is not match` را چاپ کنید.
 - در صورتی که برای این کلاس استادی از قبل تعریف شده است پیام `this class has a professor` را چاپ کنید.
 - در صورتی که هیچ کدام از حالت‌های فوق پیش نیامد پیام `professor added successfully to the class` را چاپ کنید.
- توجه کنید اگر چند خط همزمان اتفاق افتاد، پیام‌ها از بالا به پایین الویت دارند.

▼ درخواست `student_status`

`student_status <identical_num>`

در صورتی که دانشجویی با کد ملی `<identical_num>` وجود ندارد پیام `invalid student` را چاپ کنید.

در غیر این صورت در یک سطر و با یک فاصله بین آن‌ها به ترتیب نام دانشجو (`student_name`)، سال ورودی (`entering_year`)، رشته تحصیلی (`field`) و

نام (`name`) کلاس‌هایی که این دانشجو در آن‌ها شرکت می‌کند را به ترتیب ثبت نام چاپ کنید. در صورتی که این دانشجو در هیچ کلاسی ثبت نام نکرده است هیچ چیزی چاپ نکنید.

▼ درخواست `professor_status`

`professor_status <identical_num>`

در صورتی که استادی با کد ملی `<identical_num>` وجود ندارد پیام `invalid professor` را چاپ کنید.

در غیر این صورت در یک سطر و با یک فاصله بین آن‌ها به ترتیب نام استاد (`professor_name`)، رشته تحصیلی (`field`) و

نام (`name`) کلاس‌هایی که این استاد در آن‌ها تدریس می‌کند را به ترتیب ثبت نام چاپ کنید. در صورتی که این استاد تدریس هیچ کلاسی را به عهده نگرفته است هیچ چیزی چاپ نکنید.

▼ درخواست `class_status`

```
class_status <class_id>
```

در صورتی که چنین کلاسی برای این رشته وجود ندارد پیام `invalid class` را چاپ کنید.

در غیر این صورت در یک سطر و با یک فاصله بین آن‌ها به ترتیب نام (`name`) استادی که در این کلاس تدریس می‌کند را چاپ کنید. در صورتی که هیچ استادی برای تدریس این کلاس وجود ندارد کلمه `None` را به جای نام استاد، چاپ کنید.

سپس نام (`name`) تمامی دانشجویان این کلاس را به ترتیبی که در این کلاس ثبت نام کرده‌اند را در یک سطر و با فاصله چاپ کنید. اگر هیچ دانشجویی وجود ندارد هیچ چیزی چاپ نکنید.

ورودی

در هر سطر از ورودی تعدادی از دستورات گفته شده می‌آید. تضمین می‌شود که تعداد این دستورات از 200 بیشتر نخواهد بود.

برای نشان دادن پایان دستورات، بعد از آخرین سطر کلمه `end` در یک خط ورودی داده می‌شود.

نمره دهی این سوال به این صورت است:

- فاز اول 50 امتیاز
- فاز دوم 200 امتیاز

خروجی

در هر سطر از خروجی پاسخ مناسب هر دستور را چاپ کنید.

مثال

ورودی نمونه ۱

```
register_student ali 0012233445 1398 software
register_student alireza 0012233445 1397 cs
register_student erfane 0043332227 1400 software
register_professor ahmadi 0014433225 software
make_class riazijek 9876543210 math
make_class riazijek 5552255522 software
add_student 0012233444 5552255522
add_student 0012233445 5552255522
add_professor 0014433225 5552255522
student_status 0012233445
add_student 0043332227 9876543210
add_student 0043332227 5552255522
student_status 0043332227
professor_status 0014433225
class_status 5552255522
end
```

خروجی نمونه ۱

```
welcome to golestan
this identical number previously registered
welcome to golestan
welcome to golestan
class added successfully
class added successfully
invalid student
student added successfully to the class
professor added successfully to the class
```

```
ali 1398 software riaziyek
student field is not match
student added successfully to the class
erfan 1400 software riaziyek
ahmadi software riaziyek
ahmadi ali erfan
```

ورودی نمونه ۲

```
register_student karimi 0023333334 1405 math
register_professor hajjarian 9987777776 math
register_professor azari 8876666665 math
make_class adadi 6222816098 math
make_class jabr 6222816099 math
add_professor 9987777776 6222816098
add_professor 8876666665 6222816099
add_professor 9987777776 6222816099
add_student 0023333334 6222816098
add_student 0023333334 6222816099
student_status 0023333334
professor_status 9987777776
professor_status 8876666665
class_status 6222816098
class_status 6222816099
end
```

خروجی نمونه ۲

```
welcome to golestan
welcome to golestan
welcome to golestan
class added successfully
class added successfully
professor added successfully to the class
professor added successfully to the class
this class has a professor
student added successfully to the class
```


student added successfully to the class
karimi 1405 math adadi jabr
hajjarian math adadi
azari math jabr
hajjarian karimi
azari karimi

لیگ صنعتی

• محدودیت زمان: ۲ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

مینی پروژه شماره 1: لیگ صنعتی

عارف به تازگی با برنامه نویسی آشنا شده است و برای همین ذوق خاصی دارد. از این رو می‌خواهد سیستمی طراحی کند که یک لیگ فوتبال را شبیه سازی می‌کند.

در این سیستم، دستورات مختلفی وجود دارند که به شرح زیر می‌باشند:

ساختن بازیکن جدید

new player name price speed finishing defence

این دستور، بازیکنی با نام name، قیمت price، سرعت speed، قدرت تمام‌کنندگی finishing و دفاع defence می‌سازد.

توجه کنید که name از جنس رشته، و باقی پارامترها از جنس عدد می‌باشند.

هر بازیکن، یک شناسه‌ی عددی دارد که با استفاده از آن می‌توان به آن بازیکن دسترسی داشت، اولین بازیکنی که ساخته می‌شود شناسه‌ی عددی ۱، دومین بازیکنی که ساخته می‌شود شناسه‌ی عددی ۲ و ...-امین بازیکنی که ساخته می‌شود شناسه‌ی عددی i دارد.

ساختن تیم جدید

new team name money

با استفاده از این دستور، می‌توانید یک تیم جدید با نام name و پول money بسازید. در صورتی که از قبل

تیمی با اسم name وجود داشته باشد، تیم جدیدی ساخته نمی‌شود.

همچنین name از جنس رشته و money از جنس عدد می‌باشد.

هر تیم، یک شناسه‌ی عددی دارد که با استفاده از آن می‌توان به آن تیم دسترسی داشت، اولین تیمی که ساخته می‌شود شناسه‌ی عددی ۱، دومین تیمی که ساخته می‌شود شناسه‌ی عددی ۲ و i-امین تیمی که ساخته می‌شود شناسه‌ی عددی i دارد.

خرید بازیکن توسط تیم

```
buy playerID teamID
```

با استفاده از این دستور، تیم با شناسه‌ی عددی teamID بازیکن با شناسه‌ی عددی playerID را می‌خرد.

همچنین همانطور که در بالا گفته شد، هر بازیکن یک قیمت (price) دارد و هر تیمی مقداری پول money دارد و هنگام خرید یک بازیکن باید تیم مورد نظر پول کافی را برای خرید بازیکن داشته باشد و در صورتی که خرید موفقیت‌آمیز باشد، از پول تیم، به میزان قیمت بازیکن مورد نظر کم می‌شود.

توجه کنید که در امر خرید بازیکن ممکن هست حالت‌های مختلفی پیش بیاید:

- در صورتی که بازیکنی با شناسه‌ی عددی playerID وجود نداشته باشد، باید عبارت player with the id playerID doesnt exist را چاپ کنید.
- در صورتی که تیمی با شناسه‌ی عددی teamID وجود نداشته باشد، باید عبارت team with the id teamID doesnt exist را چاپ کنید.
- در صورتی که تیم مورد نظر، پول کافی برای خرید بازیکن را نداشته باشد باید عبارت the team cant afford to buy this player را چاپ کنید.
- در صورتی که بازیکن مورد نظر، در تیمی عضو باشد، باید عبارت player already has a team را چاپ کنید.

در صورتی که هیچکدام از حالت‌های بالا پیش نیایند، با موفقیت بازیکن مورد نظر خریداری می‌شود و باید

عبارت `player added to the team succesfully` را چاپ کنید.

توجه کنید که ممکن است چندین مورد از حالت‌های بالا پیش بیایند، در این صورت شما باید اولین حالتی که پیش می‌آید را در نظر گرفته و خروجی مورد نظر را چاپ کنید و از بقیه حالت‌ها صرف نظر کنید.

فروش بازیکن

```
sell playerID teamID
```

برای فروش بازیکن می‌توان از دستور بالا استفاده کرد. با استفاده از این دستور، تیم با شناسه‌ی عددی `teamID` بازیکن با شناسه‌ی عددی `playerID` را به فروش می‌رساند.

همچنین موقع فروش بازیکن ممکن است حالت‌های مختلفی پیش بیایند:

- در صورتی که تیمی با شناسه‌ی عددی `teamID` وجود نداشته باشد، باید عبارت `team doesnt exist` را چاپ کنید.
- در صورتی که تیم مورد نظر وجود داشته باشد اما بازیکنی با شناسه‌ی عددی `playerID` در آن تیم وجود نداشته باشد، باید عبارت `team doesnt have this player` را چاپ کنید.

در صورتی که حالت‌های بالا پیش نیایند، یعنی با موفقیت بازیکن مورد نظر فروخته شده است و این بازیکن دیگر به تیمی تعلق ندارد (ممکن است در آینده توسط تیمی خریداری شود). در این حالت به اندازه‌ی قیمت بازیکن فروخته شده، به پول تیم مورد نظر افزوده می‌شود. همچنین در این حالت باید عبارت `player sold succesfully` را چاپ کنید.

انجام بازی بین دو تیم

```
match teamID1 teamID2
```

با استفاده از این دستور، دو تیم با شناسه‌های عددی `teamID1` و `teamID2` با یکدیگر بازی می‌کنند.

همچنین ممکن است در اجرای این دستور، حالت‌های مختلفی پیش بیاید:

- اگر حداقل یکی از دو تیم وجود نداشته باشد، باید عبارت `team doesnt exist` را چاپ کنید.
- اگر دو تیم مورد نظر وجود داشتند، اما حداقل یکی از آنها تعداد بازیکن‌هایش کمتر از ۱۱ نفر باشد، بازی انجام نمی‌شود و باید عبارت `the game can not be held due to loss of the players` را چاپ کنید.

در صورتی‌که هیچ‌کدام از مشکلات بالا پیش نیاید، بازی با موفقیت انجام می‌شود.

در ابتدا برای توصیف روند بازی، نیاز به تعریف پارامتر قدرت برای هر تیم داریم.

قدرت تیم اول (تیم با شناسه‌ی عددی `teamID1`) برابر است با مجموع سرعت و تمام‌کنندگی ۱۱ بازیکن اول آن تیم.

همچنین قدرت تیم دوم (تیم با شناسه‌ی عددی `teamID2`) برابر است با مجموع سرعت و دفاع ۱۱ بازیکن اول آن تیم.

توجه کنید که ترتیب بازیکن‌ها در یک تیم، همان ترتیب زمانی اضافه شدن آن‌ها به تیم می‌باشد.

در صورتی که قدرت دو تیم باهم برابر نباشد، تیمی برنده‌ی بازی می‌شود که قدرت بیشتری دارد و به تعداد بردهای تیم برنده یک واحد اضافه می‌شود و به تعداد باخت‌های تیم بازنده نیز یک واحد اضافه می‌شود و به تیم برنده، به عنوان جایزه ۱۰۰۰ واحد پول داده می‌شود. در صورتی که قدرت دو تیم برابر باشد، بازی مساوی می‌شود و تعداد مساوی‌های هر دو تیم یک واحد افزایش پیدا می‌کند.

نمایش جدول تیم‌ها

rank

با استفاده از دستور بالا، باید جدول رتبه‌بندی تیم‌ها را چاپ کنید.

شما باید تیم‌ها را بر اساس تعداد بردها (هر تیمی که برد بیشتری داشته باشد رتبه بهتری دارد)، و در صورت برابری تعداد بردها، بر اساس تعداد باخت‌های کمتر مرتب کنید (اگر دو تیم بردهای برابری داشته باشند، تیم با باخت کمتر رتبه بهتری می‌گیرد) و به صورت زیر نام آن‌ها را چاپ کنید:

```

1. teamName[0]
2. teamName[1]
...

```

که در آن `teamName[0]` تیم با رتبه‌ی ۱، `teamName[1]` تیم با رتبه‌ی ۲ و... می‌باشند.

توجه کنید که تعداد مساوی‌ها مهم نمی‌باشد و همچنین اگر دو تیم تعداد باخت‌ها و بردهایشان برابر بود، تیمی که شناسه‌ی عددی کمتری دارد رتبه‌ی بهتری می‌گیرد.

پایان لیگ

```
end
```

در انتها، پس از تمام شدن لیگ، عبارت بالا در خروجی داده می‌شود و به این معنی است که ورودی‌ها تمام شده است و برنامه شما نیز پایان می‌یابد.

ورودی

ورودی شامل چندین خط می‌باشد که در هر خط یکی از دستوراتی که در صورت سوال گفته شد، داده می‌شود.

همچنین در خط آخر ورودی، دستور `end` داده می‌شود.

توجه کنید که ممکن است در ورودی داده شده، بعضی از خط‌ها خالی باشند و هیچ دستوری در آن‌ها نباشد.

خروجی

پس از اجرای برخی از دستورها، باید خروجی‌هایی را چاپ کنید که در توضیحات هر دستور در صورت نیاز گفته شده است. توجه کنید که هر خروجی باید در یک خط چاپ شود و از چاپ تمامی خروجی‌ها در یک خط پرهیز کنید.

مثال

ورودی نمونه ۱

```
new player ArshiA 1 2 3 4
new player sadegh 1 2 3 4
new player haniye 1 2 3 4
new player fateme 1 2 3 4
new player qorban 1 2 3 4
new player abbasi 1 2 3 4
new player alikmr 1 2 3 4
new player sadegh 1 2 3 4
new player hamide 1 2 3 4
new player karane 1 2 3 4
new player vanaki 1 2 3 4

new player Ars 1 100 100 100
new player sad 1 100 100 100
new player han 1 100 100 100
new player fat 1 100 100 100
new player qor 1 100 100 100
new player abb 1 100 100 100
new player ali 1 100 100 100
new player sad 1 100 100 100
new player ham 1 100 100 100
new player kar 1 100 100 100
new player van 1 100 100 100

new team t2 100
new team t1 100
buy 12 1
buy 13 1
buy 14 1
buy 15 1
buy 16 1
buy 17 1
buy 18 1
buy 19 1
```

```
buy 20 1
buy 21 1
buy 22 1

buy 1 2
buy 2 2
buy 3 2
buy 4 2
buy 5 2
buy 6 2
buy 7 2
buy 8 2
buy 9 2
buy 10 2
buy 11 2
match 1 2
rank
end
```

خروجی نمونه ۱

```
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
player added to the team succesfully
```


player added to the team succesfully

player added to the team succesfully

player added to the team succesfully

player added to the team succesfully

player added to the team succesfully

1. t2

2. t1