



هدف از این تمرین آشنایی با مفاهیم وراثت^۱ و رسیدگی به خطاها به کمک استثناها^۲ است. بنابراین طراحی و استفاده صحیح از این عناصر در این پروژه از اهمیت ویژه‌ای برخوردار است.

سیستم ثبت بازی مافیا!

در این تمرین قصد داریم تا با استفاده مهارت‌های برنامه نویسی ای که تا به حال کسب کردید، یک سیستم برای کمک به خدای بازی مافیا جهت مدیریت آن ارائه دهیم. این برنامه باید تمام اتفاقات و جریان بازی را ذخیره کند و در اعلام بعضی از نتایج و محاسبات به خدای بازی کمک کند.

توضیح بازی:

بازی مافیا از تعدادی نقش تشکیل شده است که به صورت تصادفی به بازیکنان اختصاص داده می‌شود. نقش‌هایی که در این برنامه آن‌ها را پیاده‌سازی می‌کنیم عبارت است از:

Joker, Villager, Detective, Doctor, RouinTan, Mafia, GodFather, Silencer

در بازی کلاسیک افراد به ۲ دسته مافیا^۳ و شهروند^۴ تقسیم می‌شوند. افراد از نقش یکدیگر در بازی اطلاعی ندارند و نقش‌ها محرمانه هستند. بازی در ۲ فاز روز و شب ادامه دارد. در فاز روز تمامی افرادی که هنوز زنده هستند شروع به بحث می‌کنند و سعی می‌کنند تا اعضای مافیا را پیدا کنند. پس از آن می‌توانند یکی از افرادی که به او مظنون هستند را با رای اکثریت زنده‌ها بکشند. از طرفی اعضای مافیا در فاز روز سعی می‌کنند بقیه بازیکنان را منحرف کنند تا بتوانند زنده بمانند. اما در فاز شب اعضای مافیا می‌توانند یکدیگر را بشناسند و با همدستی یکدیگر و با رای‌گیری یک نفر را کشته و از بازی حذف کنند. در فاز شب بقیه اعضای مافیا و شهروندان با توجه به نقش خاصی که دارند، عملکرهای متفاوتی دارند که در ادامه توضیح داده می‌شود.

نقش‌های شهروندان:

- کارآگاه^۵: او هر شب می‌تواند فردی را انتخاب کرده و از برنامه بپرسد که آیا این شخص مافیا است یا خیر. برنامه همیشه باید جواب درست بدهد به جز موقعی که فرد پرسیده شده پدرخوانده داشته باشد.
- پزشک^۶: او هر شب می‌تواند با انتخاب یک نفر او را نامیرا کند، به این صورت که اگر فردی که نامیرا شده است برای کشته شدن توسط مافیا انتخاب شود، نجات پیدا می‌کند و در آن شب کشته نمی‌شود. همچنین هیچ کس متوجه این که این فرد توسط مافیا برای کشته شدن انتخاب شده بوده نمی‌شود.

¹ Inheritance

² Exception

³ Mafia

⁴ Villager

⁵ Detective

⁶ Doctor

● **رویین تن:** اگر یک بار توسط مافیا برای کشته شدن در شب انتخاب شود چون یک جان! اضافه دارد، نجات پیدا کند و هیچ کس (حتی خودش) مطلع نمی شود که قرار بوده توسط مافیایا کشته شود. اما دفعه دومی که توسط مافیایا انتخاب شود جان اضافه ای نخواهد داشت و واقعا می میرد.

نقش های اعضای مافیا:

- **پدرخوانده⁷:** یکی از اعضای مافیا است که اگر کارآگاه در شب عضویت او در مافیا را از برنامه بپرسد با این که مافیا هست اما به Detective اعلام می شود که شخص مافیا نیست.
- اگر کارگاه
- **سایلنسر⁸:** او هر شب می تواند فردی را انتخاب کند و فرد انتخاب شده در طول روز بعد و رای گیری آن روز دیگر نمی تواند حرف بزند یا نظر دهد.

نقش گروه سوم:

- **جوکر⁹:** جوکر در هیچ یک از گروه های مافیا و شهروندان قرار ندارد. همچنین او قوانین برد و باخت بازی را به هم می ریزد! در صورتی که نقش جوکر در بازی وجود داشته باشد و در یک روز پس از رای گیری کشته شود، او برنده ی بازی می شود! و همه ی بازیکنان دیگر بازنده می شود.

بخش اول: ساخت یک بازی جدید

در ابتدا خدا باید با دستور Create room یک اتاق جدید ایجاد کند. هر اتاق مشخص کننده یک دور بازی است و تمام اطلاعات یک دور کامل بازی در داخل اتاق مربوط به آن نگهداری می شود. این دستور در ابتدا اسم اتاق را می گیرد و سپس به ازای هر نقش، نام و تعداد بازیکنانی که در آن نقش قرار می گیرند را دریافت می کند. نقش های موجود در این دستور می تواند هر کدام از نقش های بازی که در ابتدا معرفی شده باشد و با هر ترتیب دلخواهی ظاهر شوند.

فرمت دستور

```
Create_room <room_name> (- <role_name> <number>)*
```

مثال

```
Create_room room1 -Mafia 8 -Villager 10 -GodFather 1 -Doctor 2 -Detective 2
```

برای این دستور به نکات زیر دقت کنید:

- اگر یک نقش در آرگومان های دستور داده نشود تعداد بازیکنان آن نقش صفر در نظر گرفته می شود.
- اگر یک نقش بیش از یک بار در آرگومان های دستور بیاید، باید آخرین مقدار به عنوان تعداد بازیکنان آن نقش در نظر گرفته شود.
- مجموع اعضای خاص مافیا (شامل پدرخوانده و سایلنسر) کمتر مساوی تعداد اعضای مافیا باشد، همچنین مجموع اعضای خاص شهروندان (کارآگاه، پزشک و رویین تن) کمتر مساوی تعداد شهروندان باشد. برای مثال اگر دو نفر در هر یک از نقش های پدرخوانده و سایلنسر موجود باشند تعداد کل اعضای مافیا باید بزرگتر یا مساوی چهار نفر باشد.

Godfather⁷

Silencer⁸

Joker⁹

بخش دوم: عوض کردن بازی

خدا می تواند به صورت همزمان چند بازی را در اتاق های میز مدیریت کند. در نتیجه در هر زمان می تواند با استفاده از دستور `create room` یک اتاق جدید ایجاد کند. برای جابجایی بین اتاق های موجود می توان از دستور زیر استفاده کرد. دقت کنید که در هر لحظه از زمان اجرای برنامه، خدا می تواند هر کدام از دستورات `Create room` و `Switch room` را اجرا کند. همچنین بازی همیشه در آخرین اتاق ای که وارد آن شده است ادامه پیدا می کند. برای این دستور به نکات زیر دقت کنید:

- در صورت داده نشدن `room_name` یا غیر معتبر بودن آن باید خطای "Invalid room name" چاپ شود.
- از اجرای دستور `create room`، `room` فعلی باید به این اتاق جدید تغییر یابد و نیازی به اجرای دوباره دستور `switch room` نیست.

فرمت دستور

```
Switch_room <room_name>
```

بخش سوم: اضافه شدن کاربر به بازی

خدا باید پس از ایجاد و انتخاب اتاق مورد نظر خود، بازیکنان را با استفاده از اسم آن ها بوسیله دستور `join` اضافه کند. اسامی بازیکنان باید با فاصله از هم جدا شوند. دقت کنید تعداد آرگومان های ورودی مهم نیست و ممکن است اسامی بازیکنان در چند خط داده شوند.

فرمت دستور

```
Join (<user_name>)*
```

مثال

```
Join kamran amir
Join mahsa
Join sadaf
```

ولی در نهایت تعداد کل بازیکنان نباید از مجموع تعداد نقش هایی که در ابتدا تعریف شده بیشتر شود در صورت رخ دادن این مورد باید پیام خطای "many users" چاپ شود. هنگامی که تعداد بازیکنان دقیقاً برابر با تعداد نقش های تعریف شده شود، برنامه باید به صورت تصادفی به هر بازیکن یک نقش اختصاص دهد و خروجی را مطابق زیر چاپ کند:

```
kamran is Mafia
sadaf is Silencer
...
```

پس از این که تعداد بازیکنان لازم به بازی `join` شدند، بازی شروع می شود و مطابق روال بازی مافیا، بازی در دوره های "شب" و "روز" ادامه پیدا می کند. برای این کار در ابتدای هر روز باید پیام `Day <day_num>` و در ابتدای هر شب باید پیام `Night <night_num>` نمایش داده شود. برای مثال بعد از مشخص شدن نقش ها باید بلافاصله پیام `Day 1` نمایش داده می شود و برنامه منتظر ورودی بعدی می شود.

بخش چهارم: رای دادن در روز

در ابتدای هر روز خدا با دستور رای مشخص می‌کند که هر بازیکن به چه کسی رای داده است.

- voter_name , votee_name باید هر دو بازیکن بازی جاری باشند وگرنه باید پیام "User not joined" بدهد.
- voter_name باید اسمی به جز کسی که توسط ساینسر انتخاب شده است باشد و در غیر این صورت باید پیام خطای "This user has been silenced before" چاپ شود.
- vote نباید به کسی داده شود که قبلاً کشته شده وگرنه باید پیام خطای "User already died" چاپ شود.

فرمت دستور

```
Vote <voter_name> <votee_name>
```

مثال

```
Vote sadaf kamran
```

بخش پنجم: پایان رای‌گیری روز و اعلام نتایج

فرمت دستور

```
End_vote
```

تمام شدن رای‌ها بوسیله این دستور مشخص می‌شود. پس از این دستور برنامه باید تعداد رای‌هایی که به هر شخص داده شده را محاسبه کرده و کسی که بیشترین رای را دارد به عنوان بازیکن کشته‌شده در آن روز معرفی کند. اگر تعدادی از بازیکنان، بیشترین تعداد رای را داشتند و رای‌هایشان برابر بود به صورت تصادفی یکی از آن‌ها کشته خواهد شد. همچنین نام شخص کشته شده باید به صورت زیر چاپ شود.

```
Died <user_name>
```

پس از نمایش اسم شخص کشته شده با چاپ پیام Night <night_num> برنامه تمام‌شدن روز و شروع فاز شب را اعلام می‌کند.

بخش ششم: رای‌گیری مافیا در شب

فرمت دستور

```
Vote <voter_name> <votee_name>
```

در طول شب، همانند روز، رای‌ها بوسیله دستور Vote، جمع می‌شوند با این تفاوت که تنها اعضای مافیا (از جمله پدرخوانده و ساینسر) می‌توانند به عنوان <voter_name> انتخاب شوند در غیر این صورت باید پیام خطای "This user can not vote" چاپ شود. نحوه انتخاب شخصی که باید کشته شود هم مانند روز یعنی ابتدا برحسب رای اکثریت و اگر بین چند نفر رای‌ها برابر و حداکثر بود، به صورت تصادفی از بین این افراد یک فرد توسط مافیا کشته می‌شود. دقت کنید در طول شب دستور end_vote قابل اجرا نیست و برنامه تا زمانی که همه اشخاصی که حق رای دارند (اعضای زنده مافیا) رای نداده باشند، صبر می‌کند و سپس شخصی که توسط مافیا مورد سو قصد قرار گرفته به صورت زیر اعلام می‌شود.

Mafia try to kill <user_name>

در این بین هر نفر می تواند بیش از یک بار رای دهد و رای آخر برای او در نظر گرفته می شود. در فاز شب نیز مانند روز نمی توان به شخصی که کشته شده رای داد و باید پیام خطای "User already died" چاپ شود، همچنین voter_name , votee_name باید هر دو بازیکن بازی جاری باشند وگرنه باید پیام "User not join" نمایش داده شود.

بخش هفتم: پرس و جوی کارآگاه

فرمت دستور

Detect <detective_name> <suspect_name>

این دستور باید به تعداد کارآگاه های زنده و به ازای هر کدام ۱ بار صدا زده شود. خروجی دستور، عضو مافیا بودن فردی که نامش <suspect_name> است را چاپ کردن Yes یا No مشخص می کند و در صورتی که suspect نقش GodFather داشته باشد با این که این فرد عضو مافیا است باید جواب No چاپ شود.

- اگر دستور برای یک کارآگاه برای بار دوم صدا زده شود باید پیام خطای "Detective has already asked" چاپ شود.
- اگر دستور برای یک کارآگاه مرده یا هر نقش دیگری صدا زده شود باید پیام خطای "User can not ask" چاپ شود.
- اگر دستور برای یک فرد مرده صدا زده شود (suspect مرده باشد) باید پیام خطای "Person is dead" چاپ شود.

بخش هشتم: نجات دادن پزشک

فرمت دستور

Heal <doctor_name> <person_name>

- این دستور باید به تعداد پزشک های زنده و به ازای هر کدام ۱ بار صدا زده شود.
- اگر دستور برای یک پزشک برای بار دوم صدا زده شود باید پیام خطای "Doctor has already healed" نمایش داده شود.
 - اگر دستور برای یک پزشک مرده یا هر نقش دیگری صدا زده شود باید پیام خطای "User can not heal" چاپ شود.
 - اگر دستور برای یک فرد مرده صدا زده شود باید پیام خطای "Person is dead" نمایش داده شود.

بخش نهم: ساکت کردن سایلنسر

فرمت دستور

Silent <silencer_name> <person>

- این دستور باید به تعداد سایلنسرهای زنده و به ازای هر کدام 1 بار صدا زده شود.
- اگر دستور برای یک سایلنسر برای بار دوم صدا زده شود باید پیام خطای "Silencer has already silenced" چاپ شود.
 - اگر دستور برای یک سایلنسر مرده یا هر نقش دیگری صدا زده شود باید پیام خطای "User can not silence" چاپ شود.
 - اگر دستور برای یک فرد مرده صدا زده شود باید پیام خطای "Person is dead" نمایش داده شود.

بخش دهم: پایان شب و اعلام اتفاقات

پس از این که این سه دستور به ترتیب و هرکدام به تعداد مناسب اجرا شدند، شب به پایان رسیده و دوباره روز می شود. دقت کنید از روز دوم به بعد در ابتدای روز علاوه بر پیام Day <day_num> باید دو پیام زیر جهت نشان دادن شخص کشته شده و شخص ساکت شده نمایش داده شود.

```
Day <day_num>
Killed <user_name>
Silenced (<user_name>)*
```

نکته دیگر اینکه اگر شخصی که در شب گذشته، به وسیله مافیا مورد سوقصد قرار گرفته باشد اما توسط حداقل یکی از پزشکان نجات داده شود، دیگر نیازی به چاپ پیام کشته شدن او (پیام اول) و اگر تمام سایلنسرها در طول بازی مرده باشد، نیازی به چاپ پیام دوم نیست.

همچنین اگر یک نفر توسط دو نفر از سایلنسرها ساکت شده باشد باید تنها یک بار در لیست افراد ساکت شده ذکر شود همچنین این لیست باید به ترتیب حروف الفبا مرتب شده باشد.

توجه کنید که اگر فرد کشته شده توسط مافیایا رویین تن باشد، دکتری او را نجات نداده باشد و فرد برای اولین بار در شب کشته شده باشد، جان اضافه اش از بین می رود و در روز به عنوان فرد کشته شده نمایش داده نمی شود.

بخش یازدهم: وضعیت بازی

با این دستور در هر قسمت از بازی می توان تعداد اعضای مافیا و پلیس ها را بدست آورد:

فرمت دستور

```
Get_room_state
```

فرمت خروجی

```
Mafia = <mafia_number>
Villager = <villager_number>
```

بخش دوازدهم: پایان بازی

بازی باید تا جایی ادامه پیدا کند که یکی از دو تیم مافیا یا شهروند پیروز شوند که در یکی از دو حالت زیر رخ می دهد:

- تمام اعضای مافیا در یک اتاق کشته شوند که باید پیام Victory! چاپ شود.
- تعداد پلیس ها در هر زمان از بازی کمتر یا مساوی تعداد اعضای مافیا شود که باید پیام Loose! چاپ شود.

نکات تکمیلی:

(۱) در هر مرحله از بازی اگر اتاق عوض شد باید اطلاعات بازی در تا این مرحله نگه داری شود و بعد از بازگشت به آن اتاق روند بازی دقیقاً از همان مرحله ی قبلی از سر گرفته شود. برای مثال اگر یک بازی در حین رای گیری باشد و اتاق بازی با دستور Switch_room تغییر کند، پس از بازگشت به اتاق قبلی رای های گذشته تماماً نگه داری شده باشند و ادامه ی رای گیری انجام شود.

۲) برنامه شما باید در برابر تمامی خطاهای ممکن مقاوم باشد و در صورتی که خطایی رخ دهد با چاپ پیام مناسب رخ دادن خطا را به کاربر اطلاع دهد. همچنین توجه کنید برای رسیدگی به خطاها باید از exception استفاده کنید و استفاده از هرگونه پیاده‌سازی دیگر باعث کسر نمره خواهد شد.

۳) پیام‌های خطایی که در صورت سوال نام برده نشده، در تست اتوماتیک بررسی نمی‌شوند و این قسمت تنها در تحویل حضوری بررسی می‌شود.

نحوه‌ی تحویل

فایل‌های برنامه‌ی خود را با نام A3-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که در آن SID شماره دانشجویی شماست.

لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- این پروژه باید حتماً باید به روش شی‌گرایی و به صورت Multi File باشد و استفاده از makefile اجباری می‌باشد.
- به فرمت و نام فایل‌های خود دقت کنید.
- سوالات خود را در فروم درس مطرح کنید تا دوستانتان نیز از آن‌ها استفاده کنند.
- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.