

به نام خدا

سال نو مبارک! سال پیش این موقع که خاطرتون هست؟! اسمش عید بود، ولی بهش می گفتن دورانِ طلایی و از این چیزا! حالا هم که به سلامتی، دانشجوی یکی از بهترین دانشگاه‌های کشور هستید و نوروز رو تجربه می کنید. امیدواریم سال بسیار عالی و سرشار از خبرهای خوب داشته باشید و همیشه شاد، موفق، سلامت و درجه یک باشید و البته کرونا نگیرید! 😊



در ادامه، تمرین نوروزی در قالب سه سوال آمده است. ضمن این که کماکان به موارد ذکر شده در فایل توضیحات تمرین‌ها (موجود در سایت درس) توجه می کنید، در این تمرین، برای هر سوال یک پوشه به نام آن سوال بسازید. در هر پوشه، کل فایل‌های مربوط به آن پروژه را گذاشته و یک فایل فشرده شده در قالب zip مطابق الگوی زیر بارگذاری کنید. دقت کنید که پروژه مربوط به هر سوال از این تمرین باید در درگاه آپلود جداگانه‌ای که برای آن سوال در نظر گرفته شده، بارگذاری شود.

StudentNumber_FirstName_LastName.zip

مثال: 9031066_Ehsan_Edalat.zip

موعد تحویل هر یک از تمرین‌ها به طور جداگانه در سایت درس مشخص شده است. در صورت رعایت زمان‌های اعلام شده برای هر سوال، **نمره امتیازی** در نظر گرفته می شود.

سوال اول: تا ساعت ۲۳:۵۵ روز ۱۳ فروردین ۱۳۹۹

سوال دوم: تا ساعت ۲۳:۵۵ روز ۱۸ فروردین ۱۳۹۹

سوال سوم: تا ساعت ۲۳:۵۵ روز ۲۵ فروردین ۱۳۹۷

در صورتی که قادر به رعایت زمان‌بندی‌های توصیه‌شده نشوید، آخرین مهلت برای کسب نمره کامل سوال‌های اول و دوم تا **ساعت ۲۳:۵۵ روز ۲۰ فروردین ۱۳۹۹** و آخرین مهلت برای کسب نمره کامل سوال سوم تا **ساعت ۲۳:۵۵ روز ۲۷ فروردین ۱۳۹۹** خواهد بود. پس از زمان‌های تعیین‌شده برای هر سوال، می‌توانید تمرین خود را با احتساب ۱۵٪ کسر نمره به ازای هر روز تأخیر، تا ساعت ۲۳:۵۵ روز جمعه ۲۹ فروردین ۱۳۹۹ ارسال کنید.

* استفاده مناسب از مفاهیم و مطالب تدریس‌شده ضروری است. طراحی خوب و منطقی کلاس‌ها و اینترفیس‌ها باید متناسب با اصول برنامه‌نویسی شی‌گرا باشد. رعایت اصول پنهان‌سازی اطلاعات (information hiding)، سلسله‌مراتب مناسب ارث‌بری جهت استفاده مجدد از کدها (code reusability)، استفاده از چندریختی (polymorphism) و سایر نکات تدریس‌شده الزامی است. سعی کنید پیش از شروع پیاده‌سازی، تعداد و نام کلاس‌ها، فیلدها و متدهای مورد نیاز برنامه را تحلیل کنید و مطابق با تحلیل و طراحی انجام‌شده، برنامه را پیاده‌سازی کنید.

* رعایت اصول کدنویسی خوانا، کامنت‌گذاری و مستندسازی در قالب JavaDoc برای هر سه سوال الزامی است.

* برنامه‌ها را پیش از بارگذاری به خوبی تست و اشکال‌زدایی کنید! همه کلاس‌ها و همه متدهای موجود در برنامه‌ها کاملاً مورد بررسی و آزمون قرار بگیرند تا از درستی عملکرد برنامه اطمینان حاصل کنید. حالت‌های مختلف ورودی توسط کاربر باید بررسی شود و در صورت لزوم، پیغام خطای مناسب نمایش داده شود.

* این تمرین، در صورت مساعد بودن شرایط، تحويل حضوری خواهد داشت. تحويل حضوری در اولین جلسه کارگاه‌ها بعد از مهلت نهایی بارگذاری پروژه‌ها برگزار می‌شود و فقط پروژه‌های بارگذاری‌شده در سایت درس تحويل گرفته می‌شوند. برای رعایت مساوات بین دانشجویان گروه‌های مختلف کارگاه، امکان تغییر کدها بعد از بارگذاری تا زمان تحويل حضوری وجود ندارد.

* این تمرین برای یک‌روز کار طراحی نشده است! لطفاً در زمان‌بندی انجام تمرین دقت کافی را داشته باشید.

* در صورتی که کد هر کدام از سوالات را در اینترنت بیابید و از آن استفاده کنید، تقلب محسوب می‌شود و طبق قوانین تقلب با شما رفتار می‌شود.

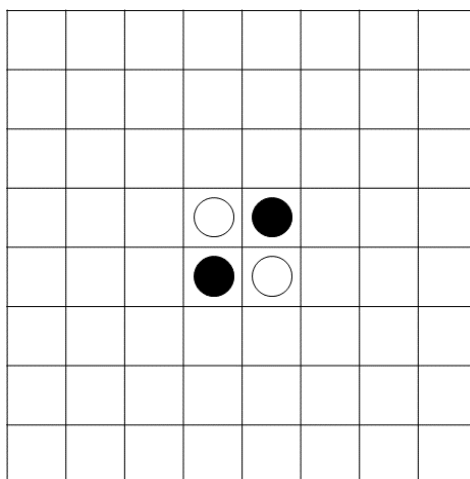
* پروژه‌های طراحی شده، به ترتیب سختی و از آسان به سخت آمده‌اند. وزن هر پروژه در نمره نهایی به ترتیب ۲۰، ۳۰ و ۵۰ از مجموع ۱۰۰ نمره خواهد بود. در نظر داشته باشید که حداکثر ۲۵٪ نمره امتیازی برای هر پروژه در نظر گرفته می‌شود و در نهایت نمره پروژه نوروز با احتساب نمره امتیازی ۱۲۵ از ۱۰۰ خواهد بود. با توجه به نمره هر پروژه و همچنین، حداکثر نمره امتیازی، در انجام پروژه‌ها برنامه‌ریزی کنید.

منتظر پروژه‌های کامل، جذاب و عالی شما هستیم! ☺

سوال اول) بازی اوتلو

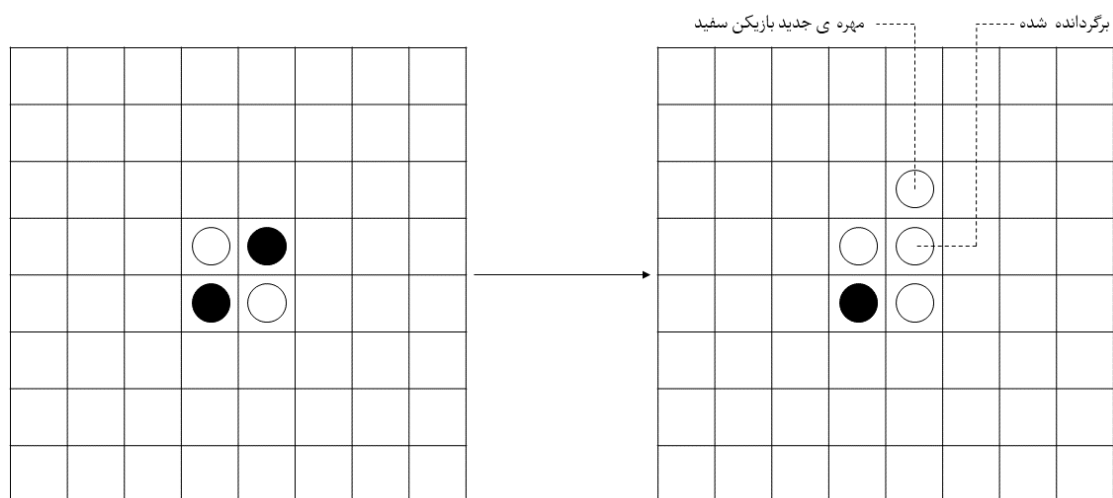
بازی اوتلو یک بازی دو نفره است که بر روی یک صفحه 8×8 انجام می‌شود. مهره‌ها هر یک دو روی سیاه و سفید دارند و هر بازیکن یک رنگ دارد. برای آشنایی بیشتر می‌توانید به نمونه‌های آنلاین مثل این [لینک](#) مراجعه کنید. توصیه می‌شود ابتدا چند بار بازی کنید تا با قوانین بازی به خوبی آشنا شوید. در هر قسمت از تعریف پروژه که ابهام وجود دارد، ملاک ما در پیاده‌سازی، قوانین بازی آنلاین است که لینک آن داده شده است.

در ابتدای بازی چهار مهره در چهار مربع مرکزی قرار می‌گیرند، به شکلی که هر مهره کنار مهره رنگ مخالف خود باشد.



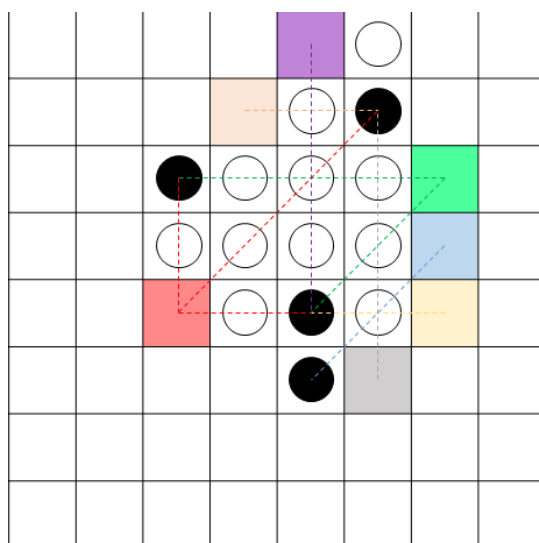
حالت اولیه

در هر نوبت، بازیکنی که نوبت او است، یک مهره به روی رنگ خود روی صفحه می‌گذارد و سعی می‌کند به شکلی این کار را انجام دهد که تعداد مهره‌های بیشتری از مهره‌های حریف گرفتار شوند تا آن‌ها را برگرداند و به رنگ مهره‌های خودش درآورد. این کار با قراردادن مهره‌ها به نحوی صورت می‌گیرد که یک خط عمودی، افقی یا اریب شکل دهند و دو سر این خط را مهره‌های بازیکن تشکیل دهد. مهره‌های حریفی که در این خط قرار می‌گیرند، برگردانده می‌شوند و تبدیل به مهره‌های بازیکن می‌شوند.

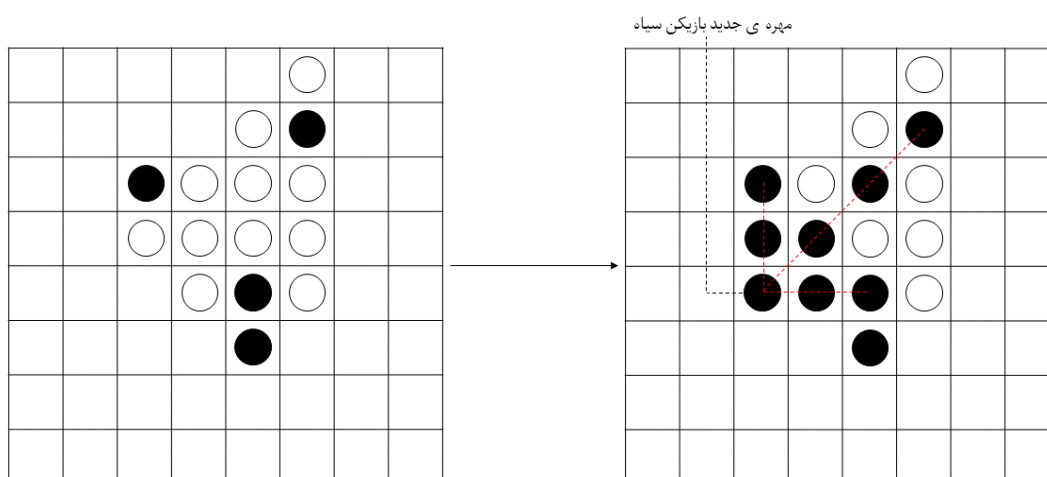


نمونه حرکت اول

توجه شود که در هر حرکت بازیکن باید مهره خود را به شکلی قرار دهد که برخی مهره‌های حریف را گرفتار کند. یعنی حتما باید مهره خود را کنار صفی از مهره‌های حریف قرار دهد که در آن سر صف یکی از مهره‌های خودش باشد.



حرکت‌های ممکن برای بازیکن سیاه و مهره‌های حریفی که با هر حرکت گرفتار می‌شوند



نمونه حرکت صحیح بازیکن سیاه

در صورتی که هیچ حرکتی برای بازیکن وجود نداشته باشد عبارت «pass» چاپ می‌شود و حریف مجدداً می‌تواند بازی کند. اگر هیچ حرکتی برای هیچ کدام از بازیکنان باقی نماند، بازی به اتمام می‌رسد. این حالت معمولاً وقتی پیش می‌آید که تمام صفحه پر شده باشد. برنده بازی کسی است که در انتهای بازی بیشترین تعداد مهره را به رنگ خود بر روی صفحه داشته باشد.

پیاده سازی

- برای پیاده سازی بازی، نمایش وضعیت صفحه در هر مرحله به شکل مناسب الزامی است. (به صورت متنی در کنسول)
- برای نمایش رنگ‌های مهره‌ها می‌توانید از دو کاراکتر (برای مثال b و w یا ۰ و ۱ و یا از [کاراکترهای یونی‌کد](#) مثل ○، ●) استفاده کنید.
- در ابتدای بازی باید چینش اولیه نمایش داده شود.
- در هر نوبت باید ذکر شود که نوبت کدام بازیکن است.
- در هر مرحله شرط خاتمه بازی بررسی شود و در صورت به اتمام رسیدن بازی برنده اعلام شود.
- در هر نوبت باید وجود داشتن حرکت برای بازیکن بررسی شود و در صورتی که حرکت ممکن نباشد، عبارت «pass» و یا عبارت مناسب دیگری چاپ شود.
- در هر نوبت، بازیکنی که نوبتش است تنها به فرمت «I C» یک رشته وارد می‌کند که I یک عدد بین ۱ تا ۸ است و C یک کاراکتر از A تا H که این رشته، خانه‌ای که مهره جدید در آن می‌خواهد قرار گیرد را نمایش می‌دهد. سپس برنامه به صورت خودکار (پس از بررسی کردن قابل قبول بودن ورودی) عملیات برگرداندن مهره‌ها را انجام می‌دهد و نتیجه را در کنسول چاپ می‌کند. ردیف‌ها و ستون‌ها به شکل زیر هستند:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |

- پیاده سازی بازی دو نفره **الزامی** است.
- پیاده سازی بازی تک نفره (انسان با کامپیوتر) **امتیازی** است. برای این بخش، برنامه تنها باید در نوبت خود به جای حریف یک رشته قابل قبول با رعایت **شروط بازی** چاپ کند و عملیات را مانند قبل انجام دهد. مقدار هوشمندی پیاده‌سازی در نمره اثرگذار است.

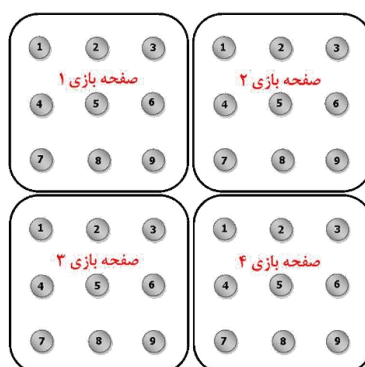
سوال دوم) پنتاگو

پنتاگو یک بردگیم است که توسط Tomas Flodén طراحی و ساخته شده است. قوانین بازی در ادامه آورده شده است. مانند شطرنج، پنتاگو دو بازیکن دارد، در این بازی هیچ حالت تصادفی یا پنهانی وجود ندارد و هدف هر بازیکن این است که بازیکن حریف در بازی ببازد (یا حداقل به حالت تساوی برسند). برای آشنایی با بازی می‌توانید به صورت [آنلاین](#) آن را بازی کنید. در هر قسمت از تعریف پروژه که ابهام وجود دارد، ملاک ما در پیاده‌سازی، قوانین بازی [آنلاین](#) است.

شروع بازی

بازی با صفحه خالی مطابق شکل زیر شروع می‌شود. هر صفحه بازی شامل ۴ صفحه بازی (بلوک) است که قابلیت چرخیدن دارند. همچنین دو دسته مهره به رنگ‌های قرمز و مشکی وجود دارند. (محدودیتی روی تعداد مهره‌ها فرض نکنید). یکی از بازیکن‌ها به تصادف یکی از این دو رنگ را دریافت کرده و بازیکن دیگر رنگ دیگر را دریافت می‌کند.

بلوک‌ها در صفحه مطابق شکل زیر شماره‌گذاری می‌شوند.



شکل ۱

تعریف‌ها

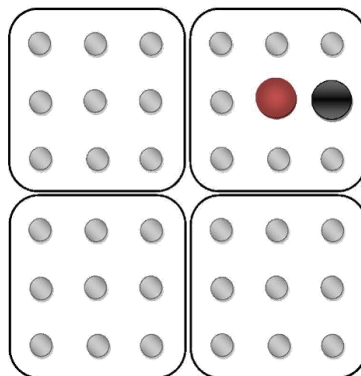
صفحه بازی جامع: کل شکل ۱ یک صفحه بازی جامع است.

صفحه بازی (بلوک): هر یک از ۴ بخش شکل ۱ یک صفحه بازی (بلوک) است و قابلیت چرخش دارد.

صفحه بازی (بلوک) خنثی: صفحه بازی که هیچ مهره‌ای در آن قرار ندارد.

توالی n تایی از مهره‌ها: n مهره هم‌رنگ، در هر جای صفحه بازی کلی، در یک ردیف عمودی، افقی یا اریب، پشت سر هم قرار گیرند.

مختصات: خانه (۰ و ۰) گوشه بالا و چپ صفحه است.



شکل ۲ - در شکل بالا مهره قرمز در در خانه (۱و۴) و مهره مشکی در خانه (۱و۵) قرار دارد.

نحوه انجام بازی

در این بازی مانند خیلی از بازی‌های دونفره دیگر، بعد از هر مرحله حرکت نوبت‌ها عوض می‌شود. هر مرحله حرکت خود شامل دو حرکت می‌شود.

۱- قرار دادن مهره‌ای با رنگ بازیکن در صفحه.

۲- چرخاندن یکی از بلوک‌ها به اندازه ۹۰ درجه در جهت ساعتگرد یا پادساعت گرد

دقت کنید بازیکن آزادانه و مستقل از اینکه مهره را در کدام بلوک قرار داده است می‌تواند انتخاب کند کدام بلوک را بچرخاند. همچنین دقت کنید در هر نوبت باید حتماً هر دو عمل انجام شود. تنها در حالتی که بلوک خنثی (خالی) در صفحه موجود باشد و یا در حالت کلی‌تر، بلوکی وجود داشته باشد که نسبت به دوران قرینه باشد می‌تواند دوران انجام ندهد. (در این حالت در واقع انگار دوران روی آن بلوک انجام شده اما تغییری در وضعیت صفحه بازی ایجاد نشده است)

برنده شدن در بازی (هدف بازی)

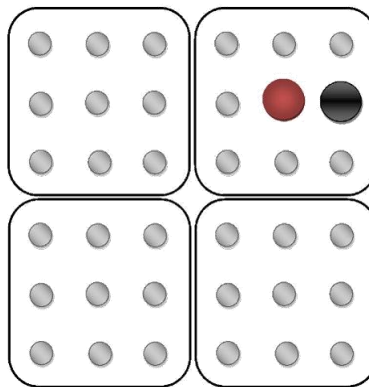
هدف بازی پنتاگو این است که تلاش کنید پیش از رقیبتان، پنج مهره خود را، در هر جای صفحه بازی کلی، در یک ردیف عمودی، افقی یا اریب، پشت سر هم قرار دهید. (یک توالی ۵ تایی بسازید).

حالت تساوی

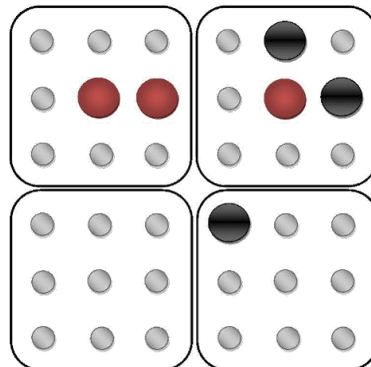
در حالتی که تمام خانه‌ها با مهره پر شوند (هیچ کدام برنده نشوند) و یا در حالتی که یک دوران هم‌زمان دو توالی ۵ تایی از در دو رنگ ایجاد کند (هم‌زمان برنده شوند)، حالت تساوی رخ خواهد داد.

توجه کنید که اگر بازیکن A با قرار دادن یک مهره برنده شود و بعد از دوران بازیکن B هم یک توالی ۵ تایی به رنگ خود بسازد این حالت یک تساوی نیست و برد بازیکن A است.

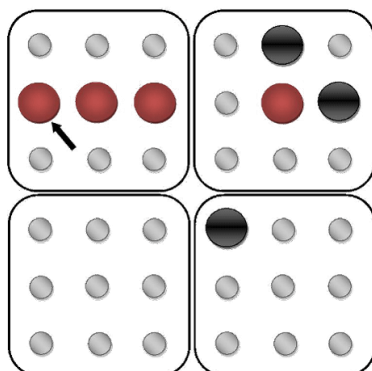
۱. مثالی از انجام بازی



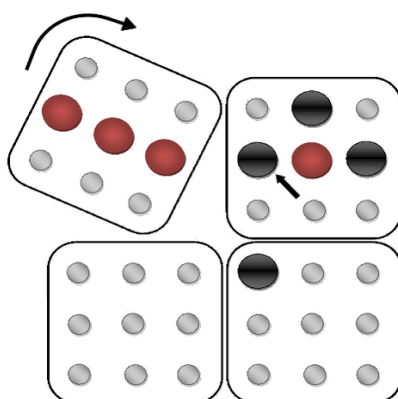
شکل ۳ - به دلیل وجود بلوک خنثی می‌توانند دوران انجام ندهند.



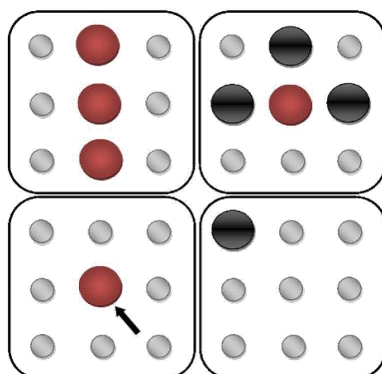
شکل ۴ - بازی ۳ مرحله پیش می‌رود... (هنوز کسی دوران انجام نداده است).



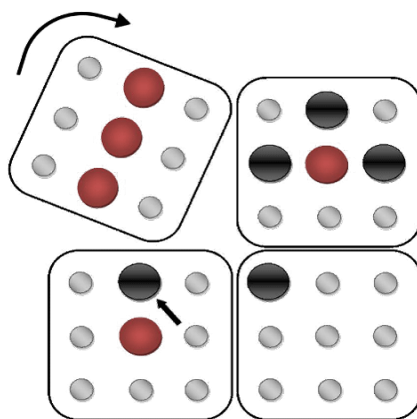
شکل ۵ - قرمز مهره خود را در مکان مشخص شده قرار می‌دهد. در این حالت قرمز سه مهره را بر روی صفحه بازی ۱ (بلوک ۱) ردیف کرده است و اگر سیاه در نوبت خود این حرکت را خنثی نکند قرمز در نوبت بعدی بازی خود پنج مهره را در یک راستا قرار می‌دهد.



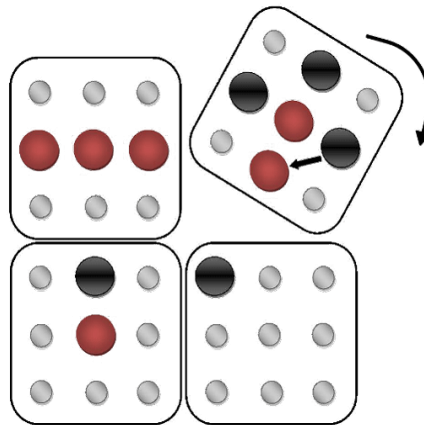
شکل ۶ - سیاه با قرار دادن مهره در بلوک شماره ۲ و چرخاندن بلوک شماره ۱ تهدید قرمز را موقتا خنثی می‌کند.



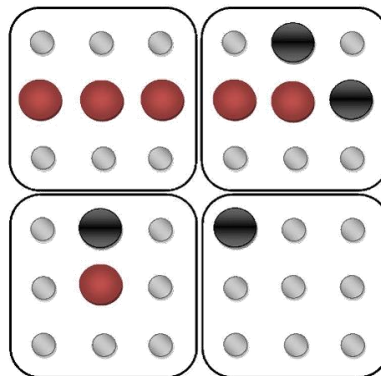
شکل ۷ - قرمز بازی را ادامه می‌دهد و مهره خود را در بلوک شماره ۳ می‌گذارد و بلوک شماره ۳ را می‌چرخاند که هنوز هم خنثی است و تاثیری در شکل بازی ندارد.



شکل ۸ - سیاه مجدداً با قرار دادن مهره در بلوک شماره ۳ و چرخاندن بلوک شماره ۱ تهدید قرمز را خنثی می‌کند.



شکل ۹ - قرمز ضربه نهایی را با چرخاندن و قرار دادن مهره در صفحه شماره ۲ می‌زند و برنده بازی می‌شود.

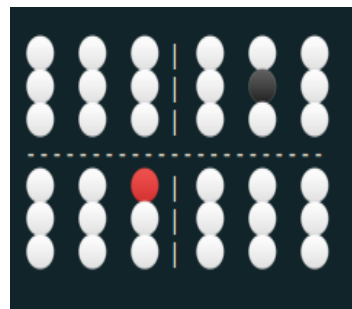
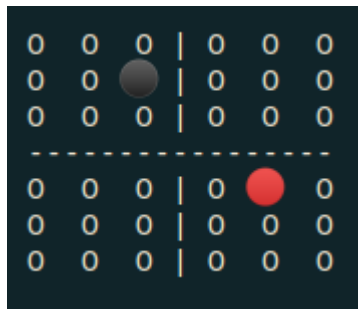
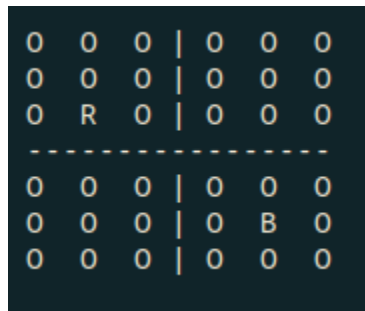


شکل ۱۰ - تصویر زیر حالت نهایی بازی قرمز در مقابل سیاه است.

پیاده سازی

طبق توضیحات داده شده بازی را پیاده سازی کنید. به نکات زیر توجه کنید.

نیازی نیست مانند مثال بالا رابط کاربری گرافیکی برای بازی خود طراحی کنید اما نیاز است که مراحل اجرای بازی حتما به صورت متنی در کنسول نمایش داده شود. باید بازی را در کنسول پیاده‌سازی کنید. برای نمایش رنگ‌ها می‌توانید از حروف الفبا انگلیسی و یا از [کاراکترهای یونی‌کد](#) (مثلا ○، ●، ◌ و یا مشابه آن) استفاده کنید یا رنگی چاپ کنید. ظاهر خوب بسته به کیفیت آن نمره امتیازی دارد. حداقل چیزی که باید پیاده‌سازی شود، نمایش درست خانه‌ها و تقسیم صفحه بازی کلی به ۴ بلوک است. در زیر چند نمونه از این نمایش‌ها را می‌بینید.



شکل ۱۱ - نمونه‌های قابل قبولی از نمایش صفحه بازی در کنسول

- ۱- تمامی قوانینی که در بالا آمده است را در پیاده‌سازی خود رعایت کنید.
 - ۲- بازی که پیاده‌سازی می‌کنید مطابق توضیحات باید در هر مرحله نوبت را عوض کند و بتواند حرکتی که کاربر می‌خواهد انجام دهد را دریافت کرده و به درستی اجرا کند.
 - ۳- در هر بخش از هر مرحله باید بررسی کنید که آیا حرکت انجام‌شده باعث برد بازیکنی شده است یا خیر. اگر یکی از بازیکن‌ها برنده شده باشد، بازی متوقف می‌شود و او به عنوان برنده معرفی می‌شود. اگر به یکی از حالات تساوی رسیدند هم این نتیجه چاپ خواهد شد.
- پیاده‌سازی بازی تک نفره (انسان با کامپیوتر) امتیازی است و باید بتواند با نوعی منطق تلاش کند بازی را برنده شود. همچنین از آنجایی که در این بازی استراتژی برد با نفر اول است، پیاده‌سازی الگوریتمی که برد نفر اول را تضمین کند نیز امتیازی است (امتیاز آن از بخش قبل جدا است).

سوال سوم) بازی هفت خبیث (UNO)

بسیاری از شما با بازی UNO (هفت خبیث) آشنایی دارید. حتما قبل انجام پروژه این بازی را چند بار انجام دهید. در این [لینک](#) می‌توانید به صورت آنلاین بازی کنید. در هر قسمت از تعریف پروژه که ابهام وجود دارد، ملاک ما در پیاده‌سازی، قوانین بازی آنلاین است. در این پروژه قصد پیاده سازی یک بازی شبیه به بازی UNO را داریم.

آماده سازی بازی:

در ابتدای بازی از کاربر تعداد بازیکنان (یک عدد بین ۳ تا ۵) را به عنوان ورودی دریافت می‌کنید. (برای توضیح بازی تعداد بازیکنان را ۴ نفر فرض می‌کنیم که یک بازیکن خود شما و ۳ بازیکن دیگر کامپیوتر خواهند بود).

در ابتدای بازی باید به هر بازیکن به صورت تصادفی ۷ کارت داده شود (انواع کارت‌ها در زیر مفصل توضیح داده شده است). پس از آن یک کارت به صورت تصادفی به عنوان کارت شروع از بین کارت‌های باقی مانده اعلام می‌شود (کارت مورد نظر باید از کارت‌های رنگ‌دار باشد. در مورد نحوه نمایش بازی در ادامه صحبت می‌کنیم) و باقی کارت‌ها باید به عنوان مخزن کارت ذخیره شود.

در ادامه باید یکی از ۴ بازیکن به عنوان شروع کننده (کاملاً تصادفی) اعلام شود.

- در هر لحظه از بازی باید جهت چرخش بازی مشخص باشد (می‌توانید بالای کنسول با عبارت clockwise یا anticlockwise جهت چرخش را مشخص کنید). در ابتدای بازی جهت چرخش ساعتگرد (clockwise) است.

بازی در یک دور اجرا می‌شود. پس از اتمام دور باید امتیاز بازیکنان نمایش داده شود. (نحوه امتیازبندی در ادامه ذکر شده است)

کارت‌های موجود در بازی

بازی از ۱۰۸ کارت تشکیل شده است.

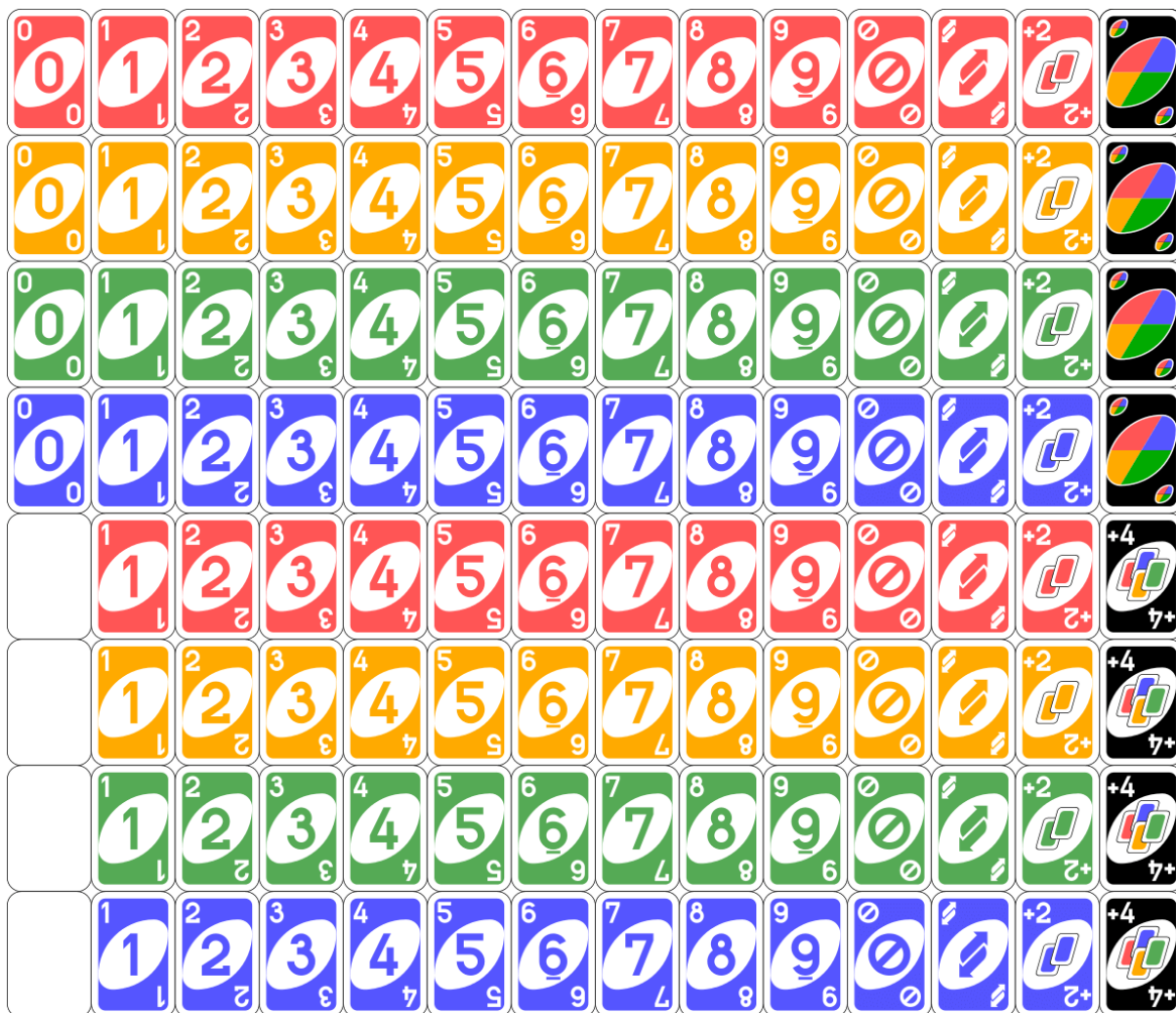
- ۲۵ کارت از نوع رنگ‌دار (قرمز، سبز، آبی، زرد) و ۸ کارت مشکی (کارت وحشی^۱).
- هر سری کارت رنگی شامل کارت‌های عددی یا حرکتی هستند.

^۱ Wild card

- اعداد شامل ۰-۹ هستند.
 - از کارت عددی زمانی می‌توان استفاده کرد که کارت روی زمین هم‌رنگ و یا هم شماره آن باشد.
- کارت‌های حرکتی شامل:
 - پرش (Skip): بازیکن بعدی نوبتش را از دست می‌دهد.
 - از کارت Skip فقط در زمانی می‌توان استفاده کرد که رنگ آن با رنگ روی زمین یکی باشد یا کارت پرش دیگری وسط باشد. همچنین اگر این کارت شروع کننده باشد، بازی از بازیکن سمت راست بازیکن اول یعنی بازیکن دوم آغاز می‌شود.
 - برگردان (Reverse): جهت بازی را باز می‌گرداند.
 - از کارت Reverse فقط زمانی می‌توان استفاده کرد که رنگ آن با رنگ کارت روی زمین یکی باشد یا کارت برگردان دیگری وسط باشد. همچنین اگر این کارت شروع کننده باشد، بازی از سمت چپ پخش کننده کارت‌ها انجام می‌شود.
- $2+ (Draw2)$: بازیکن بعدی را مجبور می‌کند که دو کارت از مخزن به صورت تصادفی بردارد و نوبت خود را از دست بدهد.
- از کارت Draw فقط در زمانی می‌توان استفاده کرد که رنگ آن با رنگ روی زمین یکی باشد یا کارت Draw دیگری در وسط باشد. در صورتی که در ابتدای بازی این کارت شروع کننده شود، نفر اول باید ۲ کارت برداشته و نوبت به نفر بعد داده می‌شود.
- کارت‌های وحشی به شرح زیر هستند:
 - رنگ وحشی: به بازیکن اجازه می‌دهد، رنگ کارت بعدی را مشخص کند.
 - وحشی $4+ (WildDraw)$: به بازیکن اجازه می‌دهد اعلام کند که کارت بعدی کدام رنگ است و بازیکن بعدی را مجبور می‌کند تا چهار کارت از مخزن به صورت تصادفی بردارد و نوبت خود را از دست بدهد.
- این کارت فقط زمانی می‌تواند روی زمین قرار داده شود که شخص نتواند با باقی کارت‌ها بازی را ادامه دهد. (مثلا اگر بر روی زمین ۷ قرمز قرار داده شده باشد و در دست آن

شخص عدد ۷ یا کارت قرمز دیگری وجود نداشته باشد، وی می‌تواند از Wild Draw استفاده کند.)

نمایی کلی از کارت‌ها به شکل زیر است:



امتیازدهی

هر کارت یک امتیاز است و امتیاز هر بازیکن برابر با مجموع امتیاز کارت‌های موجود در دست اوست.

- کارت‌های دارای عدد: امتیاز هر کارت دارای عدد برابر با عدد آن کارت است.
- کارت‌های دارای حرکت: هر کارت ۲۰ امتیاز دارد.
- کارت‌های وحشی: هر کارت ۵۰ امتیاز دارد.
- در آخر راند بازیکن با امتیاز کمتر برنده است.

توضیح یک راند:

یک راند تا زمانی ادامه پیدا می‌کند که یکی از بازیکنان کارت‌هایش تمام شود.

در زمان نوبت هر بازیکن، بازیکن مورد نظر باید سعی کند تا کارتی از کارت‌های خود انتخاب کند و به عنوان کارت وسط بگذارد. کارت انتخابی باید یکی از شرط‌های زیر را داشته باشد:

۱. با آخرین کارت وسط از لحاظ رنگ یکسان باشد.
۲. با آخرین کارت وسط از لحاظ عدد یکسان باشد.
۳. از یکی از کارت‌های حرکت‌دار (با رعایت شرط آن کارت) استفاده کند.
۴. از کارت‌های وحشی استفاده کند.

پس از آن نوبت بازیکن بعدی می‌شود. (در هر لحظه باید بگویید نوبت کدام بازیکن است)

اگر بازیکن مورد نظر قادر به انتخاب کارتی نبود (هیچ کدام از شرایط را نداشت) باید کارتی از مخزن (کاملاً تصادفی) به عنوان جریمه بردارد. سپس مجدداً به او اجازه گذاشتن کارت داده می‌شود و اگر نتواند کارتی را انتخاب کند، نوبت بازیکن بعدی می‌شود.

یکی از اتفاق‌هایی که در طول بازی ممکن است رخ دهد، استفاده از کارت‌های Draw یا Wild Draw بر روی همدیگر است. بدین صورت که مثلاً اگر شخص اول یک کارت Draw استفاده کند و شخص دوم نیز کارت Draw را بر روی زمین بیندازد، نفر سوم باید ۴ کارت بردارد (در صورتی که نمی‌تواند Draw بیاورد، ولی اگر می‌تواند، جریمه نفر بعدی ۶ می‌شود و باید ۶ کارت از مخزن به صورت تصادفی بردارد).

پیاده سازی

- در هر لحظه از زمان شما باید تعداد کارت‌های تمام بازیکنان و همچنین نوع کارت‌های خود را ببینید.
- در هر لحظه باید جهت چرخش بازی مشخص باشد.
- در هر لحظه از زمان باید کارت وسط مشخص باشد. (هر کارتی که آخرین بازیکن قرار می‌دهد کارت وسط می‌شود و کارت قبلی به انتهای مخزن کارت اضافه می‌شود).
- در انتهای بازی باید جدولی از بازیکنان با امتیاز آنها مشخص شود که بر اساس امتیاز کم به زیاد مرتب شده است.
- اگر ورودی شما غیر معتبر بود، باید پیغام خطای مناسبی نمایش داده شود.
- پیاده سازی به صورت n نفره که تمامی بازیکنان آدم باشند، نمره امتیازی دارد.

- باید از کنسول برای نمایش مراحل و ورودی گرفتن از کاربر استفاده کنید. قابل فهم بودن هر حرکت الزامی است (باید آخرین حرکت هر بازیکن قابل دیدن باشد) و زیبایی و کاربرپسند بودن پیاده‌سازی نمره اضافی دارد.

نمونه نمایش کارت‌ها

| | | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ |
| | | | | |
| 0 | +2 | reverse | skip | +4 |
| | | | | |
| \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ | \$\$\$\$\$\$\$\$\$\$\$\$\$ |

برای نمایش کارت‌ها در رنگ‌های مختلف در کنسول می‌توانید از این [لینک](#) استفاده کنید. در صورتی که سیستم ندارید و به صورت آنلاین پیاده‌سازی می‌کنید و امکان پرینت رنگی در آن محیط‌ها وجود ندارد، به جای رنگ‌های مختلف از رشته‌های مثل «R=قرمز، G=سبز، B=آبی و غیره» برای نمایش رنگ کارت‌ها در کنار نوع آنها استفاده کنید.