

## بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

**سال نو مبارک!** سال پیش این موقع که خاطرتون هست؟! اسمش عید بود، ولی بهش می‌گفتن دورانِ طلایی و از این چیزا! حالا هم که به سلامتی، دانشجوی یکی از بهترین دانشگاه‌های کشور هستید و نوروز رو تجربه می‌کنید. امیدواریم سال بسیار عالی و سرشار از خبرهای خوب داشته باشید و همیشه شاد، موفق، سلامت و درجه‌یک باشید! ☺



در ادامه، تمرین نوروزی در قالب سه سوال آماده است. ضمن این‌که کماکان به موارد ذکر شده در فایل توضیحات تمرین‌ها (موجود در سایت درس) توجه می‌کنید، 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 فشرده‌شده 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: 9031806\_Mohammad\_Ahmadpanah.zip

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

سوال اول: تا ساعت ۲۳:۵۵ روز ۴ فروردین ۱۳۹۷

سوال دوم: تا ساعت ۲۳:۵۵ روز ۹ فروردین ۱۳۹۷

سوال سوم: تا ساعت ۲۳:۵۵ روز ۱۴ 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 ۱۲ فروردین ۱۳۹۷** و آخرین مهلت برای کسب نمره کامل سوال **سوم** تا **ساعت ۲۳:۵۵ روز ۱۵ فروردین ۱۳۹۷** خواهد بود. پس از زمان‌های تعیین‌شده برای هر سوال، می‌توانید تمرین خود را با احتساب  $15\%$  کسر نمره به ازای هر روز تأخیر، تا ساعت ۲۳:۵۵ روز جمعه ۱۷ فروردین ۱۳۹۷ 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، تحویل حضوری خواهد داشت. تحویل حضوری در اولین جلسه کارگاه‌ها بعد از تعطیلات برگزار می‌شود و فقط پروژه‌های بارگذاری‌شده در سایت درس تحویل گرفته می‌شوند. برای رعایت مساوات بین دانشجویان گروه‌های مختلف کارگاه، امکان تغییر کدها بعد از بارگذاری تا زمان تحویل حضوری وجود ندارد.

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

منتظر پروژه‌های کامل، جذاب و عالی شما هستیم! ☺

## سوال اول) ماشین حساب ماتریسی

هدف از این تمرین، پیاده‌سازی اعمال ریاضی بر روی ماتریس‌ها است. برنامه به این صورت عمل می‌کند که ابتدا کاربر دو ماتریس  $X$  و  $Y$  را در ورودی تعیین می‌کند. سپس یک عبارت چندجمله‌ای درجه یک بر حسب  $X$  و  $Y$  وارد کرده و نتیجه محاسبات را از برنامه می‌گیرد.

قالب کلی عبارت چندجمله‌ای به شکل  $aX \circ bY$  است که در آن  $X$  و  $Y$  ماتریس‌های اول و دوم و  $a$  و  $b$  اعداد صحیح هستند و به جای  $\circ$ ، یکی از نمادهای  $+$  (جمع)،  $-$  (تفریق) یا  $*$  (ضرب) قرار می‌گیرد. اعداد  $a$  و  $b$  ضرایب ثابت هستند که به ترتیب بر روی ماتریس‌های  $X$  و  $Y$  اعمال می‌شوند. هر یک از ضرایب  $a$  و  $b$  می‌توانند مقادیر مثبت، منفی یا صفر داشته باشند. برای سادگی فرایند پردازش چندجمله‌ای این‌گونه در نظر بگیرید که در صورتی که مقدار ضریبی منفی باشد، حتماً آن عدد داخل پرانتز داده می‌شود.

برای یادآوری عملیات ریاضی روی ماتریس‌ها می‌توانید به [اینجا](#) مراجعه کنید.

نحوه تعریف ماتریس اول ( $X$ ) و ماتریس دوم ( $Y$ ) به این صورت است که اندازه ماتریس‌ها از ابتدا تعیین نمی‌شود؛ بلکه کاربر هر سطر از آن ماتریس را وارد می‌کند و پس از آن با وارد کردن یک خط خالی، اتمام تعریف یک ماتریس را مشخص می‌کند. درایه‌های هر سطر از ماتریس با ویرگول (,) از یکدیگر جدا می‌شوند.

در زیر نمونه‌ای از نحوه اجرای این برنامه را مشاهده می‌کنید:

Define the first matrix (X):

0,1,2

1,1,1

Define the second matrix (Y):

0,0,0

-1,-2,-3

Enter your polynomial expression:

$2X+(-3)Y$

Result:

0,2,4

5,8,11

Enter your polynomial expression:

$(-2)Y$

Result:

0,0,0

2,4,6

به قواعد مربوط به عملیات جمع (یا تفریق) و ضرب ماتریس‌ها توجه داشته باشید. در صورتی که اندازه ماتریس‌های ورودی سازگار با عملیات درخواستی نباشد، برنامه شما باید پیغام خطای متناسبی به کاربر نمایش دهد.

+ پیاده‌سازی برنامه به طوری که کاربر بتواند بین عملوندها و عمل‌گرها هر تعداد فاصله دلخواه قرار دهد (مثال:  $5Y * 4X$ )، نمره امتیازی دارد.

+ پیاده‌سازی برنامه به طوری که الزامی به وجود پرانتز برای مشخص کردن اعداد منفی وجود نداشته باشد (مثال:  $-2X-4Y$ )، نمره امتیازی دارد.

## سوال دوم) نبرد ناوهای جنگی

هدف از این تمرین، طراحی و پیاده‌سازی یک بازی ساده دونفره مشابه بازی معروف کاغذی نبرد ناوهای جنگی است. پس از این‌که به اندازه کافی در [اینجا](#) بازی را انجام دادید و با آن آشنا شدید، به پیاده‌سازی این سوال بپردازید. رابط کاربری بازی از طریق کنسول است و نیازی به پیاده‌سازی گرافیکی نیست!

قوانین بازی به شرح زیر است:

هر بازیکن یک جدول  $10 \times 10$  در اختیار دارد که می‌تواند ۵ ناو جنگی در آن قرار دهد. هر ناو دارای عرض ۱ واحد و طولی به دلخواه از مقادیر ۲، ۳، ۴ یا ۵ است. در ابتدای بازی، ناوها باید به طور عمودی یا افقی در صفحه چیده شوند. پس از این‌که هر یک از بازیکن‌ها ناوهای خود را بر روی صفحه خود قرار داد، بازی آغاز می‌شود. در این لحظه، بازیکن از موقعیت ناوهای حریف اطلاعی ندارد.

بازیکنان به نوبت به صفحه حریف تیراندازی می‌کنند. این کار با اعلام مختصات در صفحه حریف صورت می‌گیرد. اگر در آن واحد از صفحه، قطعه‌ای از ناو حریف موجود باشد، به آن ضربه وارد می‌شود و در صورتی که تمامی قطعات یک ناو ضربه بخورند، آن ناو نابود می‌شود. این کار تا زمانی ادامه پیدا می‌کند که یک بازیکن همه ناوهای حریف خود را نابود کند. در این صورت، بازیکن به عنوان برنده اعلام می‌شود.

در ابتدای بازی، بازیکن می‌تواند از بین تیر معمولی یا تیر خطادار یکی را انتخاب کند. برخلاف تیر معمولی که با تعیین مختصات، دقیقاً به همان واحد از صفحه شلیک می‌شود، در صورت انتخاب تیر خطادار، ممکن است به جای مختصات مشخص‌شده، یکی از ۸ واحد همسایه آن شلیک شود!

X		
	X	

نحوه عملکرد تیر خطادار

واحد انتخاب‌شده برای شلیک X است اما به واحد X شلیک می‌شود! اگرچه در بعضی موارد ممکن است به همان واحد انتخاب‌شده نیز شلیک شود.

	X	

نحوه عملکرد تیر معمولی

دقیقاً به همان واحدی که انتخاب شده است، شلیک می‌شود.

تمامی مراحل بازی توسط کنسول انجام می‌شود. در ابتدای بازی، محیط بازی باید امکان **مشاهده صفحه** و انتخاب نقاط مورد نظر برای قراردادن ناوها را با واردکردن مختصات داشته باشد. در هر مرحله از بازی باید بازیکن بتواند نمایی از صفحه خود و نمایی از صفحه حریف را مشاهده کند. در نمای صفحه خود باید محل ناوها و شلیک‌های انجام‌شده توسط حریف مشخص باشد. از طرفی، در نمای صفحه حریف باید نقاطی که تا به حال به آنها شلیک کرده به همراه موفقیت یا عدم

موفقیت آن شلیک نمایش داده شود. واضح است که بازیکن نباید قادر به مشاهده محل قرارگیری ناوهای حریف باشد.

علاوه بر حالت بازی دونفره، برنامه شما باید امکان انجام یک بازی کامل را با کامپیوتر فراهم کند. کامپیوتر باید بتواند حداقل به نقاطی از صفحه که قبلاً به آنها حمله نکرده است، به صورت تصادفی (random) حمله کند. در ابتدای شروع برنامه، باید از کاربر پرسیده شود که مایل به انجام بازی با کامپیوتر است یا می‌خواهد بازی را دونفره انجام دهد.

وارد کردن اعداد نامعتبر باید با پیام خطای مناسب به کاربر اعلام شود و درخواست گرفتن ورودی مجدداً صورت بگیرد. ناوها نیز باید در مختصات معتبر؛ یعنی در محدوده صفحه و بدون اشتراک بین آنها قرار داده شوند.

یک نمای نمونه قابل مشاهده برای یکی از بازیکنان (شامل صفحه بازیکن و صفحه حریف) می‌تواند به شکل زیر باشد. از نمادهای مختلف برای نمایش قطعات سالم ناوها (@) و قطعات تخریب‌شده ناوها (#) در صفحه بازیکن، و شلیک موفق (&) و شلیک ناموفق (X) در صفحه حریف استفاده کنید.

	0	1	2	3	4	5	6	7	8	9
0					X					
1										
2							X			
3										
4				&	&	&				
5				X					&	
6									&	
7										
8										
9										

صفحه حریف

	0	1	2	3	4	5	6	7	8	9
0										
1			@	@	@	@				
2										
3		@					@	@	@	
4		@								
5										
6		#		#	#	#	#	#		
7		@								
8		#								
9										

صفحه بازیکن

+ پیاده‌سازی بازی به طوری که در چاپ صفحه بازیکن و صفحه حریف به جای نمادهای بالا، از رنگ‌های آبی برای نمایش قطعات سالم ناوها، قرمز برای نمایش قطعات تخریب‌شده ناوها، سبز برای نمایش شلیک موفق و زرد برای نمایش شلیک ناموفق استفاده شود، نمره امتیازی دارد.

+ پیاده‌سازی بازی به طوری که کامپیوتر به صورت تصادفی شلیک نکند و دارای نوعی هوش مصنوعی ساده باشد، نمره امتیازی دارد.

## سوال سوم) بازی کارتی حیوانات جنگل

هدف از این تمرین، طراحی و پیاده‌سازی یک بازی کارتی به نام حیوانات جنگل است. مشابه دو سوال قبلی، این برنامه نیز تحت رابط کاربری متنی (کنسول) است. این بازی دونفره است و در ابتدای بازی، کارتهایی از حیوانات مختلف جنگل به طور تصادفی (random) در اختیار هر یک از بازیکنان قرار داده می‌شود. به هر بازیکن می‌توان از یک نوع حیوان بین صفر تا پنج کارت داده شود به طوری که در مجموع ۳۰ کارت در اختیار بازیکن قرار می‌گیرد. سپس بازیکن ۱۰ کارت از بین آن‌ها را انتخاب کرده و بازی را با آن‌ها شروع می‌کند. پس از این مرحله، بازیکن تنها به کارتهای انتخاب‌شده دسترسی خواهد داشت.

در هر نوبت از بازی، بازیکن علاوه بر مشاهده کارتهای خود، کارتهای بازیکن حریف را نیز می‌تواند مشاهده کند و استراتژی مناسبی برای ادامه بازی در نظر می‌گیرد. قوانین بازی به این شرح است:

هر کارت حیوان یک بازیکن می‌تواند به یکی از حیواناتهای حریف آسیب بزند و در صورتی که حیوانی قدرت کافی برای آسیب‌زدن را نداشته باشد، می‌تواند با همکاری یک یا چند حیوان دیگر این کار را انجام دهد. هر حیوان دارای انرژی و جان است که در هر بار استفاده از آن حیوان، بسته به نوع آسیب، مقدار مشخصی از انرژی آن کاسته و به همان میزان، از جان حیوان حریف کم می‌شود. در صورتی که بیش از یک حیوان در آسیب‌زدن نقش داشته باشند، کاهش انرژی به طور یکسان بین همه آن‌ها تقسیم می‌شود.

دقت کنید که قبل از اعمال آسیب‌زدن، باید بررسی شود که حیوان(های) انتخاب‌شده برای آسیب انرژی کافی را دارند؛ یعنی مقدار انرژی باقی‌مانده برای حیوان(های) انتخاب‌شده بعد از آسیب‌زدن، مقدار کمتر از صفر نمی‌شود. در غیر این صورت، عمل آسیب‌زدن صورت نمی‌گیرد و بازیکن باید مجدداً انتخاب دیگری را انجام دهد. در صورتی که میزان آسیب مساوی یا بیش از جان حیوان حریف باشد، حیوان حریف می‌میرد و آن حیوان از لیست کارتهای حریف حذف می‌شود.

همچنین، هر یک از بازیکنان می‌توانند در نوبت خود، به جای آسیب‌زدن به حریف، انرژی یکی از حیوانات زنده خود (که هنوز جان دارد) را به طور کامل ترمیم کند. تجدید انرژی برای هر بازیکن نهایتاً ۳ بار قابل انجام است.

پایان بازی زمانی اعلام می‌شود که یکی از بازیکنان تمامی حیواناتهای حریف را از بین برده باشد و آن بازیکن برنده بازی است.



علاوه بر حالت بازی دونفره، این برنامه باید قابلیت انجام بازی با کامپیوتر را نیز داشته باشد. انتخاب کارتها در ابتدای بازی و گزینش حیوان، نوع آسیب‌زدن یا تجدید انرژی در هر نوبت از بازی تصادفی انجام می‌شود.

هر حیوان یک یا دو روش برای آسیب‌زدن به حیوان حریف دارد که در جدول زیر مشخص شده است:

نماد حیوان	نام حیوان	روش آسیب‌زدن ۱ (میزان آسیب)	روش آسیب‌زدن ۲ (میزان آسیب)	انرژی	جان
	شیر	زخمی‌کردن (۱۵۰)	کشتن (۵۰۰)	۱۰۰۰	۹۰۰
	خرس	زخمی‌کردن (۱۳۰)	کشتن (۶۰۰)	۹۰۰	۸۵۰
	ببر	زخمی‌کردن (۱۲۰)	کشتن (۶۵۰)	۸۵۰	۸۵۰
	کرکس	زخمی‌کردن (۱۰۰)	-	۶۰۰	۳۵۰
	روباه	زخمی‌کردن (۹۰)	-	۶۰۰	۴۰۰
	فیل	صدمه‌زدن (۷۰)	حمله‌کردن (۵۰)	۵۰۰	۱۲۰۰
	گرگ	کشتن (۷۰۰)	-	۷۰۰	۴۵۰
	گراز	صدمه‌زدن (۸۰)	-	۵۰۰	۱۱۰۰
	اسب آبی	حمله‌کردن (۱۱۰)	-	۳۶۰	۱۰۰۰
	گاو	حمله‌کردن (۹۰)	زخمی‌کردن (۱۰۰)	۴۰۰	۷۵۰
	خرگوش	گازگرفتن (۸۰)	-	۳۵۰	۲۰۰
	لاک‌پشت	گازگرفتن (۲۰۰)	-	۲۳۰	۳۵۰

در ادامه برای توضیح بهتر چگونگی بازی، یک نمونه از سناریوی بازی آورده شده است:

کارت‌های انتخاب‌شده توسط بازیکن اول از بین ۳۰ کارت تصادفی اولیه: ۲ شیر، ۳ ببر، ۱ فیل، ۱ اسب آبی، ۲ گراز، ۱ لاک‌پشت

کارت‌های انتخاب‌شده توسط بازیکن دوم از بین ۳۰ کارت تصادفی اولیه: ۵ خرس، ۳ کرکس، ۱ خرگوش، ۱ لاک‌پشت

نوبت بازیکن اول: انتخاب یک شیر برای کشتن یکی از خرس‌های حریف (پس از این انتخاب، انرژی شیر بازیکن به  $500 = 1000 - 500$  و جان خرس آسیب‌دیده حریف به  $350 = 500 - 150$  تبدیل می‌شود. زیرا که میزان آسیب کشتن توسط یک شیر، ۵۰۰ واحد است).

نوبت بازیکن دوم: انتخاب همکارانه یک خرگوش و یک لاک‌پشت برای گازگرفتن اسب آبی حریف (پس از این انتخاب، انرژی خرگوش به  $210 = 350 - 140$  و انرژی لاک‌پشت به  $90 = 230 - 140$  تبدیل می‌شود و جان اسب آبی بازیکن اول به  $720 = 1000 - 280$  تغییر می‌کند. زیرا که مجموع میزان آسیب گازگرفتن خرگوش و لاک‌پشت ۲۸۰ است که کاهش انرژی به طور یکسان بین هر دو تقسیم می‌شود و از هر یک  $140 = 280 / 2$  واحد کاسته می‌شود).

نوبت بازیکن اول: با انتخاب تجدید انرژی برای شیری که در نوبت قبلی به حیوان حریف آسیب زده است، انرژی آن شیر به مقدار کامل (۱۰۰۰) تبدیل می‌شود.

نوبت بازیکن دوم: بازی به همین ترتیب ادامه می‌یابد...

+ پیاده‌سازی بازی به طوری که کامپیوتر به صورت تصادفی کارت‌های اولیه و حرکات خود را انتخاب نکند و دارای نوعی هوش مصنوعی ساده باشد، نمره امتیازی دارد.