



به نام خدا

## دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران مبانی کامپیوتر و برنامه‌نویسی



استاد : دکتر هاشمی

پروژه ۲

نیمسال اول ۹۸-۹۹

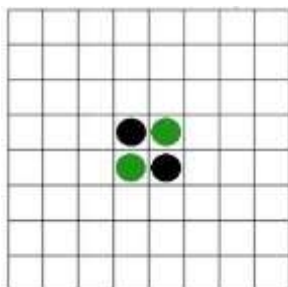
### Othello

بازی اتللو در رده اول بازی های فکری دنیا است که از تعدادی مهره با دو وجه رنگی و یک صفحه بازی تشکیل شده است.

این بازی به صورت دو نفره انجام می شود. شما قرار است این بازی را در دو بخش پیاده سازی کنید.

#### توضیح بازی

در ابتدا بازی با دو مهره سیاه و دو مهره سفید آغاز می شود.



بازیکن صاحب مهره های سیاه بازی را آغاز می کند. هر یک از دو بازیکن به نوبت یک حرکت انجام می دهند. هر بازیکن فقط می تواند مهره اش را در جایی قرار دهد که یک یا چند مهره حریف را محاصره کند. انجام حرکت به معنی گذاشتن یک مهره (همرنگ بازیکن) در صفحه و محصور کردن یک یا چند مهره حریف در یک یا چند راستا است (عمودی، افقی و یا قطری). در نتیجه مهره های محاصره شده را برگردانده و به رنگ مهره خود در آورید. این به معنی محاصره و تصاحب مهره های حریف و تغییر آن ها به مهره ی هم رنگ بازیکن است.

در انتهای بازی، بازیکنی که صاحب مهره های بیشتری باشد، به عنوان برنده انتخاب می شود.

## توضیح پروژه

### بخش اول :

**طراحی بالا به پایین :** طراحی بالا به پایین یک روند است که از یک مفهوم سطح بالا به سمت سطح پایین تر حرکت می کند. مفهوم طرح سطح بالا، بزرگتر، گسترده تر و عمومی تر است. طرح سطح پایین کوچکتر، خاص منظره تر و دقیق تر است.

طراحی بالا به پایین در هر رفتار انسانی رخ می دهد و مربوط به سیستم درک و حافظه در مغز انسان می باشد. برای مثال یک نقاش را در نظر بگیرید که برای به وجود آوردن آنچه که در ذهنش دارد ابتدا یک طرح کلی می کشد ( سطح بالا ) و در نهایت با اضافه کردن نکات و ریزه کاری ها در بخش های مختلف طرح آن را کامل می کند. طراحی بالا به پایین در برنامه نویسی کامپیوتر نیز همین گونه می باشد. با یک تعریف کلی از صورت مسئله شروع می شود و به نیازهایی که باید توسط آن برنامه رفع شود می پردازد. این طرح سطح بالا به صورت مکرر شکسته می شود و به نیازها و طرح های سطح پایین تر تبدیل می شود تا به اندازه کافی ساده برای پیاده سازی در یک زبان برنامه نویسی باشد. شما باید در این بخش طراحی بالا به پایین خود و در واقع شکستن مسئله به مسائل کوچکتر را تحویل دهید. قالب این طراحی بر عهده خودتان می باشد و باید به گونه ای باشد که شیوه ی تقسیم بندی مسئله تان واضح باشد.

### بخش دوم :

در این بخش شما باید این بازی را بدون نمایش گرافیکی پیاده سازی کنید. بازی با نوبت بازیکن اول که b نامیده می شود، شروع می شود. بازیکن با دادن یک شماره (که نشان دهنده ی خانه ی انتخاب شده است)، محل قرارگیری مهره را مشخص می کند و در صورتی که قرار دادن مهره منجر تغییر بقیه ی مهره های بازی شد، تغییرات لازم در صفحه ی بازی ایجاد شود. سپس نوبت بازی عوض شده و به همین منوال بازی ادامه پیدا میکند تا تمام محل های قرارگیری مهره ها پر شوند، و یا حرکتی باقی نمانده باشد. در پایان، برنده ی بازی با نمایش پیامی مناسب مشخص می شود.

هر مرحله از بازی روند بازی به شکل زیر می باشد.

## ۱. نمایش نقشه‌ی بازی

در هر مرحله نقشه‌ی بازی که در ابتدای بازی از ۸ ردیف و ۸ ستون تشکیل شده است باید نمایش داده شود. برای نمایش نقشه شما باید به جای خانه‌های خالی از . ، به جای خانه‌های پر شده با مهره های سیاه از b و به جای خانه‌های پر شده با مهره های سفید از w استفاده کنید. توجه شود که در هر مرحله تعداد مهره های سفید و مشکی صفحه را نشان دهید.

	1	2	3	4	5	6	7	8 [b=2 w=2]
10	.	.	.	.	.	.	.	.
20	.	.	.	.	.	.	.	.
30	.	.	.	.	.	.	.	.
40	.	.	.	w	b	.	.	.
50	.	.	.	b	w	.	.	.
60	.	.	.	.	.	.	.	.
70	.	.	.	.	.	.	.	.
80	.	.	.	.	.	.	.	.

ورودی خروجی ۱ : نمایش نقشه بازی

## ۲. گرفتن مختصات

در هر مرحله از بازی با توجه به نوبت بازی شما باید مختصات خطی که بازیکن قصد انتخاب آن را دارد به عنوان ورودی بگیرید. این با مختصات را با یک عدد نشان می‌دهیم. با توجه به نقشه‌ی بازی، به هریک از خانه‌های جدول، عددی بین ۱۱ تا ۸۸ نسبت داده می‌شود. برای مثال، عدد ۲۵ نشان دهنده‌ی خانه واقع در سطر دوم و ستون پنجم است.

در این پروژه، شما باید با دادن پیام مناسب به کاربر و گرفتن یک عدد به عنوان ورودی، خانه‌ی مورد نظر را پیدا کنید.

BLACK player's turn :

در قسمت ورودی برنامه شما در صورتی که مختصات صحیحی وارد نشد باید این موضوع با نمایش پیامی مناسب اطلاع داده شود و دوباره از بازیکن مختصات را به عنوان ورودی بگیرید. همچنین اگر بازیکن مختصاتی که قبلاً در آن مهره‌ای قرار داده شده را وارد کرد، باید با نمایش پیامی مناسب اطلاع داده شود و دوباره از او مختصات را به عنوان ورودی بگیرید.

بعد از گرفتن مختصات باید مهره‌ای که متعلق به بازیکن است، در این مختصات قرار بگیرد.

- در صورتی که مختصاتی که بازیکن انتخاب می‌کند غیرمجاز باشد، باید این موضوع با نمایش پیامی مناسب اطلاع داده شود و دوباره از بازیکن مختصات را به عنوان ورودی بگیرید.
- در صورتی که مختصات انتخاب شده مجاز بود، باید تغییرات لازم در صفحه‌ی بازی اعمال شود و سپس، صفحه‌ی بازی جدید دوباره نمایش داده‌شود.

### ۳. پایان بازی

در هر مرحله از بازی، شما موظف هستید که صفحه بازی را به درستی چک کنید. حالتی وجود دارد که در آن ها به هیچ یک از بازیکن ها اجازه‌ی انجام حرکت بیشتر داده نمی‌شود و یا تمامی خانه‌های جدول، پر شده‌است. در این حالات بازی باید به اتمام رسیده‌است و باید خاتمه یابد.

### ۴. نمایش نتایج

در پایان بازی برنامه شما باید امتیاز هر بازیکن و برنده‌ی بازی را نمایش دهد.

```
The WHITE player has won!  
Black: 25  
White: 39
```

ورودی خروجی ۲ : خروجی نمونه برای اعلام نتایج نهایی

## چند نکته و راهنمایی

- بهتر است که از یک حلقه‌ی کنترلی کلی برای تکرار مراحل کلی بازی استفاده کنید.
  - می‌توانید برای وضعیت هر کدام از خانه‌ها از یک آرایه‌ی دوبعدی کمک بگیرید.
  - توجه داشته باشید که شما باید در این پروژه توابع خود را در فایل‌های جداگانه با هدر فایل‌های مخصوص خود بنویسید.
- شیوه‌ی تقسیم‌بندی توابع به خودتان مربوط است و هر روش قابل قبولی پذیرفتنی است.

## نحوه‌ی نمره‌دهی :

عنوان	نمره
تقسیم درست مسئله به مسائل کوچکتر	۱۵
نام‌گذاری مناسب و اصولی متغیرها	۵
استفاده از تمام ورودی‌های تابع در آن	۵
نام‌گذاری مناسب توابع	۵
عدم وجود قطعه کد تکراری	۱۰
استفاده‌ی درست از هدر فایل‌ها برای تقسیم بندی توابع	۵
دریافت صحیح ورودی	۵
استفاده از آرایه‌ی دوبعدی	۵
رسیدگی به خطاها	۵
چاپ صحیح نقشه‌ی بازی	۵
تغییر درست نوبت‌ها با توجه به شرایط بازی و خطاها	۵
آپدیت درست بازی	۱۰
کامنت‌گذاری مناسب در هرجایی که نیاز به مستندسازی دارد	۵
تست و اجرای برنامه	۱۵

## نحوه‌ی تحویل پروژه :

شما باید فایل‌های h ، c و همچنین طراحی بالا به پایین خود را در قالب pdf در قسمت در نظر گرفته شده با عنوان CA2P1[Upload here] در صفحه‌ی درس در سایت CECM آپلود نمایید. مهلت آپلود تا ساعت ۲۳:۵۵ شنبه ۱۰م آذر است . تأخیر در سه روز نخست به ازای هر روز ۱۰ درصد و در سه روز دوم به ازای هر روز ۱۵ درصد جریمه خواهد داشت. پس از این شش روز، به هیچ وجه نمره‌ای در نظر گرفته نخواهد شد. همچنین توجه کنید که با توجه به تأخیر در نظر گرفته شده به هیچ وجه امکان تمدید پروژه‌ها وجود ندارد.

فایل‌های خود را در یک فایل زیپ با فرمت “zip” و با نام CA2P1\_SID.zip قرار دهید که SID همان شماره‌ی دانشجویی شماست. برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۰۰۰ باشد، باید نام فایل خود را CA2P1\_810198000.zip قرار دهید.

پروژه‌ها برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی میشوند و انجام آنها به صورت انفرادی خواهد بود. همچنین، در صورت شباهت میان دو پروژه (که به وسیله‌ی نرم افزارهای مربوطه چک میشود) برای هر دو نفر نمره‌ی صفر در نظر گرفته خواهد شد .

در صورت وجود هرگونه سوال میتوانید پرسشهای خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید یا به [icsp98.ca@gmail.com](mailto:icsp98.ca@gmail.com) ایمیل بزنید.

موفق و سربلند باشید