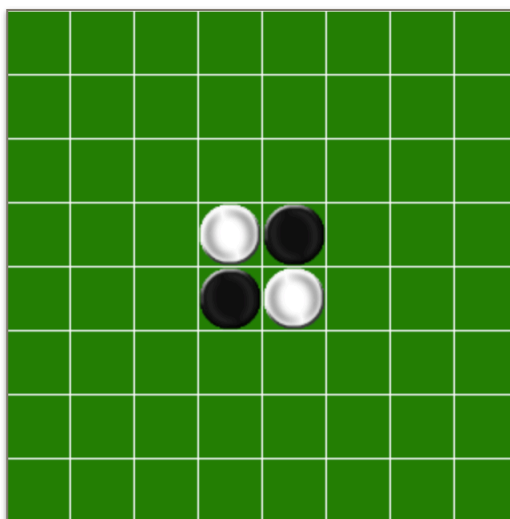


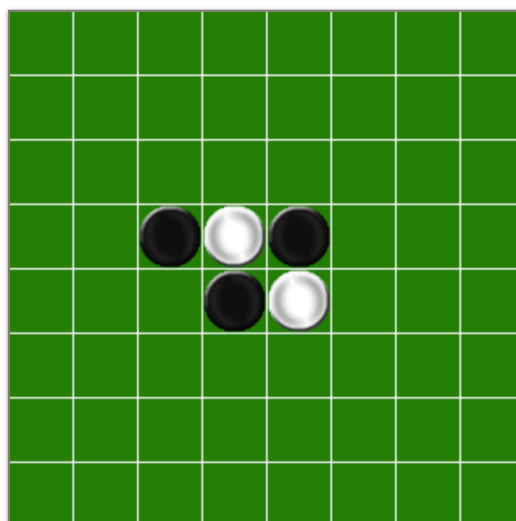
پروژه پایانی مبانی برنامه سازی

توضیح بازی:

اتللو (Othello) نام رسمی یک بازی است که با نام ریورسی (Reversi) نیز شناخته می‌شود. این بازی دوفره در یک صفحه‌ی شطرنجی 8 در 8 مانند صفحه شطرنج انجام می‌شود. هدف بازی اتللو در اختیار گرفتن تعداد بیشتری از خانه‌ها نسبت به حریف است. شروع بازی اتللو به شکل زیر است:

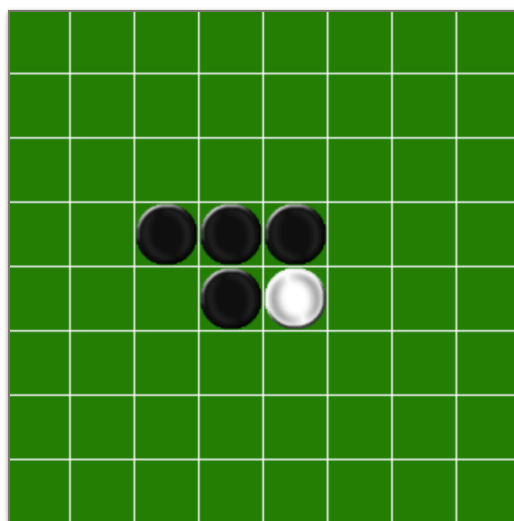


سیاه اول بازی می‌کند. در بازی اتللو تنها یک قانون وجود دارد، پرش. شما می‌توانید از روی چند مهره ی پیوسته ی حریف پرش انجام دهید. این پرش در مسیر سطر، ستون و قطر انجام می‌شود. البته با انجام هر پرش از روی مهره های حریف، مهره ی خودی در جای خود باقی می‌ماند. نکته ی دیگر این که همه ی مهره های حریف که در مسیر پرش هستند، تبدیل به مهره ی خودی می‌شوند. به شکل زیر نگاه کنید که اولین حرکت را در یک بازی نشان می‌دهد.

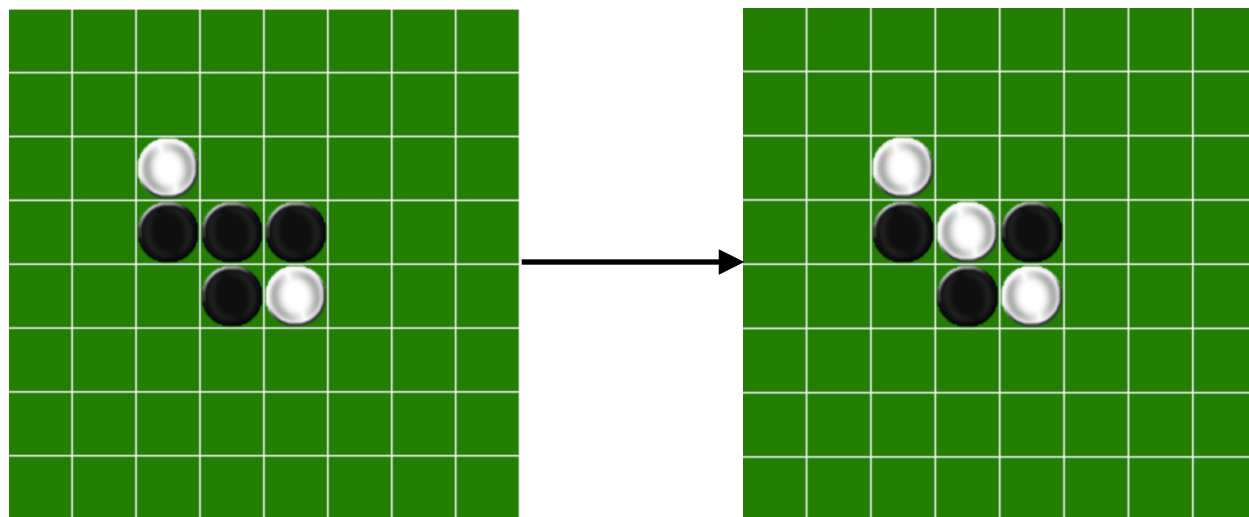


مهره ی E4 سیاه از روی مهره ی D4 سفید پرش انجام داده است. همان طور که می بینید مهره ی پرش کننده در جای خود باقی می ماند.

اما حرکت هنوز تمام نشده است. مهره ی D4 سفید به سیاه تغییر رنگ می دهد.

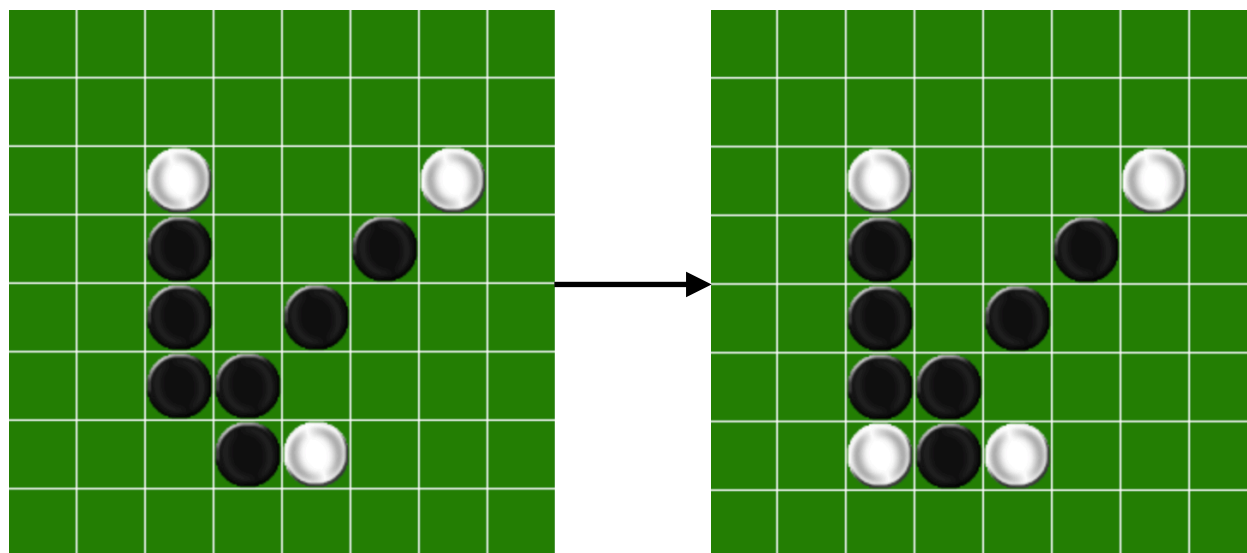


حال حرکت بعدی را در نظر بگیرید.

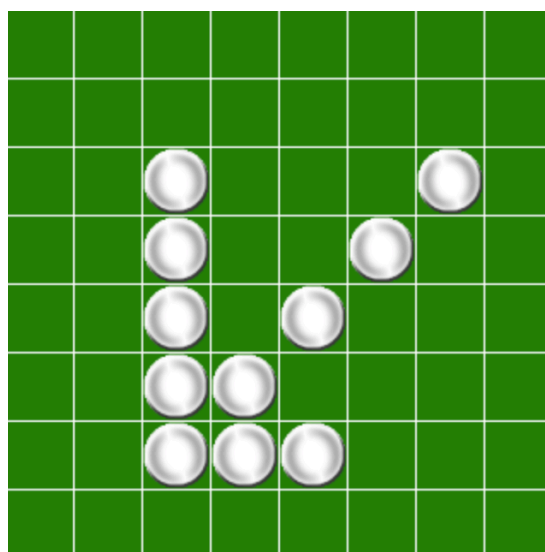


مهره ی E5 سفید از روی مهره ی D4 سیاه پرش می کند. باز هم می بینید که مهره ی E5 در جای خود باقی می ماند.

میتوانیم پرش های از چند جهت به طور همزمان داشته باشیم. شکل زیر را ببینید.



بازیکن سفید در محل مشخص شده با فلش، مهره‌ای قرار می‌دهد. حال می‌توان پرش مورد نظر سفید را از سه جهت (با سه مهره‌ی سفید موجود) در نظر گرفت. در این حالت همه‌ی پرش‌های ممکن را در نظر می‌گیریم و همه‌ی مهره‌های سیاه موجود در مسیر این پرش ها تغییر رنگ می‌دهند. بنابراین با انجام این پرش، بازی به حالت زیر درمی آید.



قوانین بازی:

- مهره را جایی قرار دهید که یک یا چند مهره حریف را محاصره کند. یعنی مهره قرار داده شده تعدادی از مهره های حریف را بین مهره جدید و مهره هم رنگ شما در صفحه محاصره کند سپس مهره های محاصره شده را چرخانده و به رنگ مهره خود درآورید. این به معنی محاصره و تصاحب است.
- قرار دادن مهره در صفحه به صورت نوبتی انجام می گیرد.
- مهره جدید را فقط در محلی می توان قرار داد که مهره ای از حریف محاصره شود. به عبارتی مهره ای به صورت آزاد در صفحه نمی تواند وجود داشته باشد و همه مهره ها حتما در مجاورت مهره های دیگر هستند.
- خط محاصره می تواند افقی یا عمودی یا مورب یا هر سه باشد
- فقط مهره هایی را می توان تصاحب کرد که بین مهره جدید و مهره قبلی موجود در صفحه محاصره شده باشند
- در صورتی یکی از بازیکنان در نوبت خود، مکانی برای محاصره حریف نداشته باشد و نتواند حتی یک مهره او را محاصره کند، در این صورت حرکت به حریف واگذار می شود. تا زمانی که امکان محاصره برایش ایجاد شود.
- در صورتی یکی از بازیکنان در نوبت خود، ۳ بار پشت سر هم مکانی اشتباه را برای گذاشتن مهره انتخاب کند، در این صورت حرکت او نادیده گرفته شده و حرکت به حریف واگذار می شود.
- اگر انتخاب حرکت بعدی بیشتر از ۱ ثانیه طول بکشد در این صورت حرکت او نادیده گرفته شده و حرکت به حریف واگذار می شود.
- در پایان بازیکنی که تعداد مهره های بیشتری روی صفحه بازی داشته باشد، برنده است.
- مهره هایی که در جریان بازی در بین مهره های شما قرار گرفته اند به رنگ مهره های شما تبدیل نمی شوند و فقط مهره هایی که در راستا های 8گانه مهره هایی که قرار دادید به رنگ مهره های شما در می آیند.

نکات پیاده سازی:

۱- زمین بازی فعلی را از طریق آرگومان های تابع main دریافت کنید برای اینکار به کد زیر دقت کنید:

```
#include<stdio.h>
int main(int argc,char* argv[])
{
    int counter;
    printf("Program Name Is: %s",argv[0]);
    if(argc==1)
        printf("\nNo Extra Command Line Argument Passed Other Than Program Name");
    if(argc>=2){
        printf("\nNumber Of Arguments Passed: %d",argc);
        printf("\n----Following Are The Command Line Arguments Passed----");
        for(counter=0;counter<argc;counter++){
            printf("\nargv[%d]: %s",counter,argv[counter]);
        }
        return 0;
    }
```

```
}
```

حال اگر در ترمینال برنامه بالا را کامپایل و اجرا میکنیم:

```
$ gcc test1.c
```

```
$ ./a.out
```

```
Program Name Is: ./a.out
```

```
No Extra Command Line Argument Passed Other Than Program Name
```

اگر جلوی a.out یک سری ورودی وارد کنیم خروجی به صورت زیر خواهد بود.

```
./a.out first second third
```

```
Program Name Is: ./a.out
```

```
Number Of Arguments Passed: 4
```

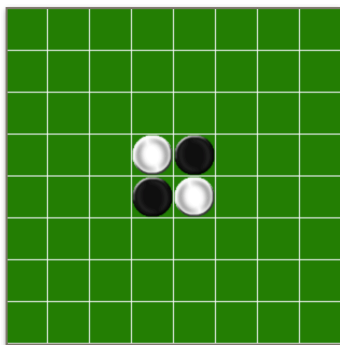
```
----Following Are The Command Line Arguments Passed----
```

```
argv[0]: ./a.out
```

```
argv[1]: first
```

```
argv[2]: second
```

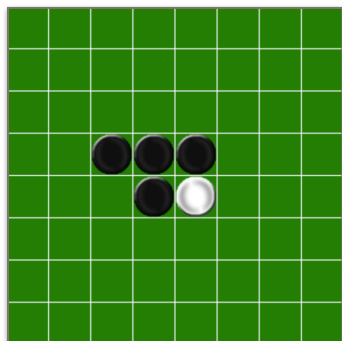
```
argv[3]: third
```



به طور مثال فرم ارسال زمین بازی روبه‌رو به آرگومان به صورت زیر میباشد:

```
./a.out 00000000 00000000 00000000 00021000 00012000 00000000 00000000 00000000 1
```

به جای مهره سیاه عدد ۱ و به جای مهره سفید عدد ۲ (حتما با همین روال زمین بازی را از آرگومان دریافت کنید!) متغیر آخر نشان دهنده رنگ بازیکن (شما) میباشد.



۲- کدی که مینویسید باید **تنها و تنها** حرکت بعدی نسبت به زمین بازی فعلی را چاپ (printf) کند. یعنی برنامه‌ی شما بایستی **تنها یک بار** زمین بازی را از آرگومان دریافت کند و با توجه به همان زمین حرکت بعدی (به صورت x,y) را printf کند سپس از برنامه خارج شود. (از چاپ هر خط اضافی در برنامه پرهیز کنید!)

برای عکس روبه رو مقدار زیر باید چاپ شود.

۳- در صورتی که حرکت چاپ شده غیر مجاز باشد (با توجه قوانین بالا) و یا مدت زمان زیادی طول بکشد (۱ ثانیه) تا حرکت چاپ شود. نوبت به بازیکن روبه‌رو داده میشود.

۴- برای تست برنامه خود در مقابل هوش مصنوعی برنامه خود را کامپایل کرده سپس فایل باینری (0.*) را به برنامه مصحح دهید. سپس با انتخاب الگوریتم مورد نظر دکمه start را بزنید. نتیجه بازی کد شما با هوش مصنوعی نمایش داده خواهد میشود.

پروژه به صورت گروه های دو نفره میباشد. هر دو نفر بایستی در پیاده سازی پروژه به یک اندازه شریک باشند به گونه که وظایف زیر را بین خود تقسیم کنند.

۱- ورودی گرفتن از آرگومان و پردازش اولیه جدول (نفر اول)

۲- چک کردن مجاز بودن هر حرکت و ساختن لیست حرکات مجاز (نفر دوم)

۳- الگوریتم پیدا کردن حرکت مناسب بعدی (هر دو نفر)

موارد تصحیح پروژه:

کارایی -> ۴۰ نمره (باید بتواند ۴ الگوریتم هوش مصنوعی را شکست دهد) - **نمره مشترک بین هر دو نفر**
تسلط -> ۴۰ نمره (باید هر یک به قسمتی از کاری که انجام داده تسلط کامل داشته باشد) - **نمره مجزا بین دو نفر**
تمیزی و رعایت اصول -> ۲۰ نمره (نام گذاری صحیح، دندانه گذاری، تابع نویسی صحیح و استفاده از پوینتر برای جابه‌جا ارسال متغیر ها به جای تعریف global) - **نمره مشترک بین دو نفر**

موارد امتیازی:

استفاده از git -> ۱۵ نمره (بایستی هر نفر در branch جدا کد زده باشد و یک branch که merge شده این دو شاخه است علاوه بر آن دو شاخه تحویل داده شود)
شرکت در tournament -> ۲۵ نمره (در صورتی که جزو ۳ تیم اول شود)

موارد رد شدن پروژه (نمره صفر):

در صورتی که یکی از اعضای تیم هیچگونه تسلطی بر روی پروژه نداشته باشد (پروژه گروهی انفرادی انجام شده باشد).
در صورتی که از تیم های دیگر تقلب شده باشد.