

شرح پروژه

در این پروژه می‌خواهیم یک محیط مجازی پویا را شبیه‌سازی کنیم. در ابتدا به شرح قوانین حاکم بر این محیط می‌پردازیم.

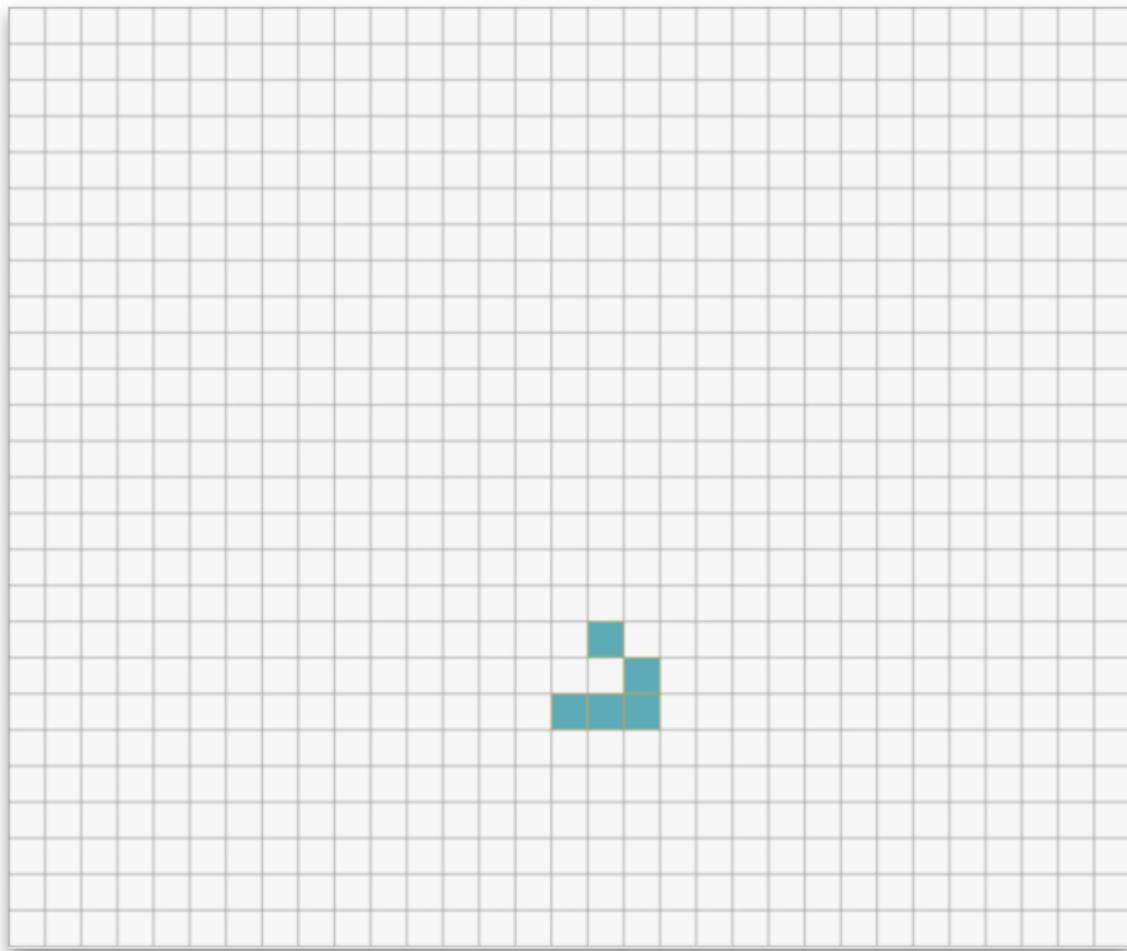
شرح عالم هیروت

زمان: زمان را در این دنیا به صورت گسسته فرض کنید. می‌خواهیم برنامه شما از لحظه صفر شبیه‌سازی را شروع کند و تا بی‌نهایت کار خود را ادامه دهد.

نقشه عالم: این دنیا به صورت یک مستطیل مشبک $n * m$ است که وضعیت هر سلول آن یا زنده است یا مرده. هر سلول با ۸ سلول همسایه خود به شکل زیر تعامل دارد:

- یک سلول زنده که کمتر از ۲ سلول زنده در همسایگی خود داشته باشد، می‌میرد.
- یک سلول زنده که بیش از ۳ سلول زنده در همسایگی داشته باشد، می‌میرد.
- یک سلول زنده که دقیقاً ۲ یا ۳ همسایه زنده داشته باشد، به بقای خود ادامه می‌دهد.
- یک سلول مرده که دقیقاً ۳ سلول زنده در همسایه‌هایش داشته باشد، به زندگی بازمی‌گردد.
- برای سلول‌های کناره نقشه (ردیف بالا و پایین، ستون چپ و راست)، همسایه‌هایی که وجود ندارند را هم‌ارز سلول مرده غیرقابل تغییر در نظر بگیرید.

یک نمونه از نقشه عالم با ۵ سلول زنده (باقی سلول‌های داخل نقشه مرده محسوب می‌شوند):



۱ سلول و ۸ سلول همسایه آن را در تصویر زیر مشاهده می‌کنید:

A	B	C
H	Cell	D
G	F	E

اگر برنامه‌ای بنویسیم که در هر لحظه از زمان، قوانین بالا را روی دنیای ما اعمال کند، فعالیتی شبیه شکل زیر رو خواهیم دید:



جزئیات شبیه‌سازی

- شبیه‌سازی با زمان $t = 1$ شروع می‌شود و تا بی‌نهایت پیش می‌رود.
- در هر t باید صفحه خروجی را کاملاً پاک کنید (می‌توانید از راهنمایی‌ای که در ادامه آمده استفاده کنید) و نقشه همان لحظه را در خروجی نمایش دهید.
- برای نمایش سلول زنده از کاراکتر ■ و برای نمایش سلول مرده از □ استفاده کنید.

تابعی برای پاک کردن خروجی

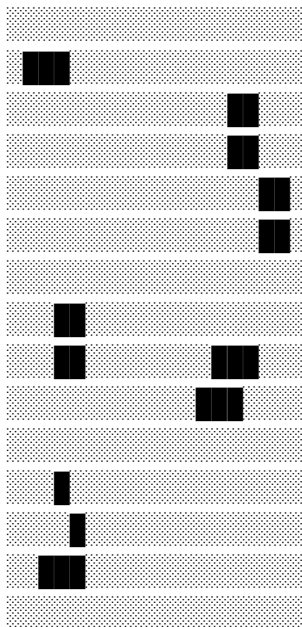
```
1 | def clear():
2 |     os.system('cls') if os.name == "nt" else os.system('clear')
```

ورودی و خروجی

برای تست پروژه شما فایل به نام `map.init` در کنار برنامه شما قرار داده می‌شود. اطلاعات داخل این فایل نقشه اولیه جهان را مشخص می‌کند. برنامه شما باید شبیه‌سازی را از این نقشه شروع کند و ادامه دهد.

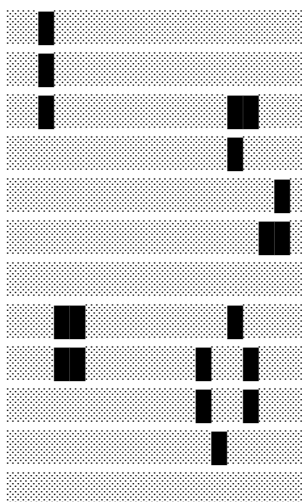
خروجی برنامه شما در هر لحظه باید نقشه حاصل از اعمال قواعد در آن لحظه باشد. برای مثال در زمان $t = 1$ باید نقشه‌ای را در خروجی چاپ کنید که حاصل از اعمال تغییرات لازم روی نقشه داخل `map.init` است.

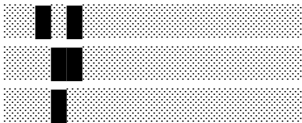
فایل `map.init` نمونه



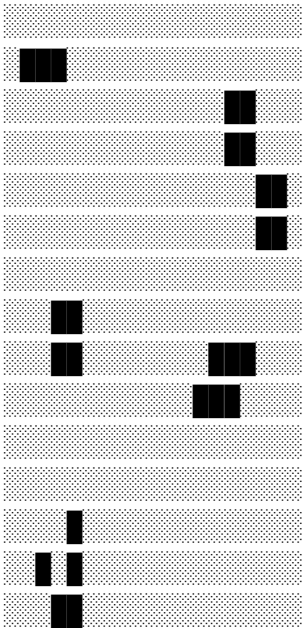
خروجی نمونه

در $t = 1$:

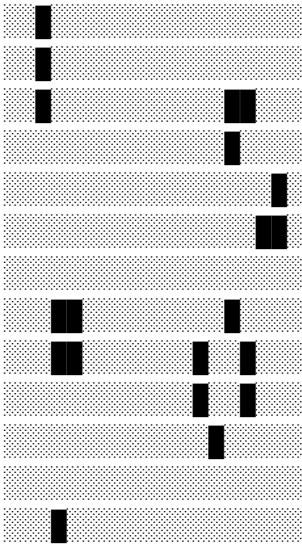




در $t = 2$:



در $t = 3$:





قواعد نمره‌دهی

- کپی کردن کد (حتی بخشی کوچک) به هر شکل (چه از اینترنت و چه از دوستان) پذیرفته نیست.
- هیچ قطعه کدی نباید خارج از تابع باشد (به جز `import` ها و خط مربوط به صدا زدن تابع اصلی برنامه).
- حین تحویل پروژه از شما خواسته می‌شود که برنامه خود را با نقشه ورودی (`map.init`) دیگری اجرا کنید، پس حتما برنامه خود را با انواع مختلف ورودی‌ها تست کنید.
- پس از تحویل پروژه باید در یک ارائه شفاهی کد و روند کدنویسی خود را توضیح دهید. در صورتی که نمره این بخش را نگیرید، نمره کل پروژه صفر می‌شود.

بخش‌های امتیازی

- تعریف توابع مناسب، استفاده از نام‌گذاری معنی‌دار و به‌طور کلی تمیزی کد نمره امتیازی دارد (امتیاز بیشتر، حداکثر تا ۱۰ درصد نمره کل پروژه).
- پیاده‌سازی با استفاده از اصول OOP نمره امتیازی دارد (با توجه به کیفیت طراحی کلاس‌های شما، حداکثر تا ۱۰ درصد نمره کل پروژه).