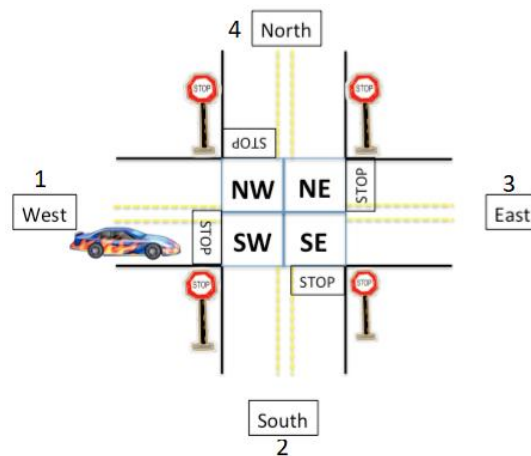


## صورت پروژه

هدف از این تمرین، آشنایی شما با مفاهیم **threading** و همچنین نحوه ی استفاده از سمافور در زبان های برنامه نویسی می باشد. مساله بدین شکل می باشد که ما چهارراهی به شکل زیر داریم.



ماشین ها در این چهارراه، از ۴ سمت می توانند وارد و یا خارج شوند. بنابراین محیط وسط چهارراه را میتوان به ۴ بخشی که در شکل بالا میبینید، تقسیم کرد. فرض کنید یک ماشین بخواهد از جنوب به غرب رود، در این حالت، باید مناطق **se, ne, nw** را قفل کند، تا بتواند با امنیت از این مسیر عبور کند. پر واضح است که اگر یک ماشین بخواهد از شمال به جنوب برود و در همان لحظه ماشینی دیگر بخواهد از جنوب به شمال برود، باید بتوانند به طور همزمان از این چهارراه عبور کنند، ولی ماشینی که میخواهد از غرب به شرق یا برعکس برود، نمیتواند به صورت همزمان از چهارراه عبور کند.

فرضیات مساله:

- همه ی ماشین ها هنگامی که به چهارراه میرسند، به مدت یک ثانیه توقف میکنند.
- یک ماشین تنها در صورتی به چهارراه وارد می شود که تمام مناطقی که میخواهد از آنها عبور کند، آزاد باشد.
- در صورتی که یک ماشین، نتواند به همه ی مناطق مورد نیازش دسترسی داشته باشد، به مدت ۲ ثانیه توقف خواهد کرد و سپس مجددا وضعیت راه ها را بررسی خواهد کرد.
- مدت زمانی که طول میکشد تا یک ماشین از چهارراه عبور کند، به تعداد مناطقی که میگذرد، بستگی دارد، مثلا اگر از شرق به شمال برود، یک ثانیه و از شرق به جنوب برود، ۳ ثانیه زمان خواهد برد.

## ورودی برنامه

ورودی مساله، یک سری مسیر است که مسیر ماشین ها را مشخص میکند. مثلا در شکل زیر مسیرها به ترتیب از غرب به شرق، شرق به غرب، غرب به شمال و .... می باشند.

```
jobsList = [  
    car(1,3),  
    car(3,1),  
    car(1,4),  
    car(2,4),  
    car(3,1)]  
await asyncio.wait(jobsList)
```

## خروجی برنامه

خروجی برنامه باید ترتیب عبور ماشین ها از چهارراه را نمایش دهد. همانطور که در شکل زیر مشاهده میکنید، از آنجایی که ماشین ها به صورت همزمان میخواهند از چهارراه عبور کنند، ضروری است که خروجی های برنامه، در اجراهای متفاوت، فرق کند(به دلیل رخداد raceCondition بین تردها).

```
\OS-Proj1  
λ python38 cars.py  
car with [3 to 1] wanna pass the intersection.  
car with [1 to 3] wanna pass the intersection.  
car with [3 to 1] passes safely  
car with [1 to 3] passes safely  
car with [1 to 4] wanna pass the intersection.  
car with [1 to 4] passes safely  
car with [2 to 4] wanna pass the intersection.  
car with [2 to 4] passes safely  
car with [3 to 1] wanna pass the intersection.  
car with [3 to 1] passes safely  
All Workers Completed  
  
\OS-Proj1  
λ python38 cars.py  
car with [2 to 4] wanna pass the intersection.  
car with [2 to 4] passes safely  
car with [3 to 1] wanna pass the intersection.  
car with [1 to 3] wanna pass the intersection.  
car with [3 to 1] passes safely  
car with [1 to 3] passes safely  
car with [3 to 1] wanna pass the intersection.  
car with [3 to 1] passes safely  
car with [1 to 4] wanna pass the intersection.  
car with [1 to 4] passes safely  
All Workers Completed
```

- این پروژه تحویل حضوری دارد و گرفتن نمره ی کامل، منوط به حضور در جلسه ی تحویل است که زمان آن متعاقبا اعلام خواهد شد. در این جلسه شما باید کدی که در کورس ویر آپلود کردید را بر روی رایانه ی خود، مجددا کامپایل کرده و از برنامه خروجی بگیرید. همچنین باید بر روی کد مسلط باشید. دقت داشته باشید که عدم شرکت در این جلسه، موجب از دست دادن درصد زیادی از نمره خواهد شد.
- برای برنامه نویسی از هر زبانی که دوست داشته باشید، میتوانید استفاده کنید ولی در صورت استفاده از ماژول `asyncio` در زبان پایتون به همراه مستند سازی دقیق، ۱۵ درصد نمره ی اضافی تعلق خواهد گرفت.
- مستند سازی برنامه، ۱۰ درصد نمره ی اضافی خواهد داشت.
- مشورت با دوستان بلامانع است ولی در صورت تقلب و استفاده از کد دیگران، در صورت تشخیص موجب کسب نمره ی -۱۰٪ برای طرفین خواهد شد.
- سورس برنامه ی خود را به همراه فایل اجرایی آن، به صورت zip شده با فرمت `os_proj1_{student#}_{name}.zip` بر روی کورس ویر آپلود کنید.
- در صورت داشتن هرگونه سوال مرتبط با این پروژه، می توانید با ایمیل [sina.tamjidy@gmail.com](mailto:sina.tamjidy@gmail.com) در ارتباط باشید.

با آرزوی موفقیت و سربلندی

سینا تمجیدی