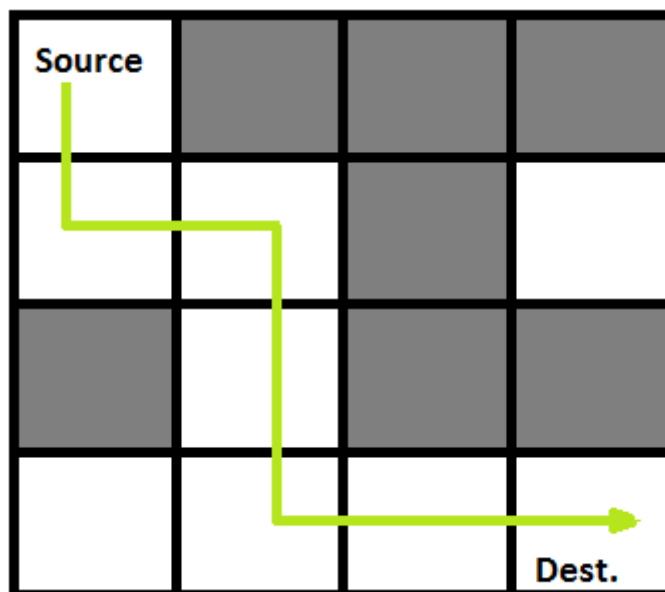


موش و پنیر (مسئله ماز "Maze")



در این مسئله از شما خواسته می شود که در یک ماز از یک خانه‌ی شروع در هشت جهت ممکن حرکت کنید تا به خانه‌ی مقصد در ماز برسید.
درواقع شما باید بین دو نقطه را با الگوریتم عقبگرد (Backtracking) و به کمک پشته (Stack) مسیریابی کنید.



همانگونه که در عکس مشاهده میکنید ماز یک ماتریس $N * N$ می باشد که خانهی مبدا $maze[0][0]$ و خانهی مقصد $maze[N - 1][N - 1]$ قرار دارد.

در این ماتریس خانه هایی که داخل آنها 0 وجود دارد مسیر و آنهایی که 1 اند دیوار است و قابل تردد نمی باشد.

موش شما فقط در هشت جهت شرق، جنوب، غرب و شمال و شمال شرقی، شمال غربی، جنوب شرقی و جنوب غربی میتواند حرکت کند.

مواردی که باید در نظر بگیرید (از 100 نمره)

۱. پیاده سازی پشته (25 نمره)

۲. خواندن ورودی از فایل (maze به صورت 0 و 1 ای) (15 نمره)

۳. پیدا کردن مسیر (25 نمره)

۴. چاپ کردن مسیر (25 نمره)

۵. محاسبه مرتبه زمانی (10 نمره)

مواردی که نمره اضافه دارند (50 نمره اکسترا بر 100)

۱. تولید ماز رندوم دارای مسیر با سایز مشخص شده، ماز رندوم باید مسیر داشته باشد (10 نمره)

۲. پیاده سازی گرافیکی (10 نمره)

۳. پیدا کردن تعداد کل مسیر ها (10 نمره)

۴. نوآوری و خلاقیت و تمیزی کد (10 نمره)

۵. شروع کردن و پایان یافتن مسیر حرکت موش براساس ورودی دلخواه کاربر (10 نمره)

نحوه تحویل پروژه:

- باید گروه های دو نفری یا تک نفری شده باشید. (ترجیحا دو نفری، برای اینکه بخواهید تک نفری ارائه بدین باید دلیل داشته باشید و با تی ای ها هماهنگ کنید)
- شما باید بعد از ثبت کردن کد خود توسط نماینده گروه در کوئرا، باید در جلسه ی حضوری از کد خود دفاع کنید.

- فقط و فقط نماینده گروه باید فایل های پروژه (سورس کد) را در کوئرا آپلود کند.
- اگر دفاع نکنید نمره ای به گروه شما تعلق نخواهد گرفت.
- در مورد تاریخ و ساعت دفاع به شما بعد از پایان ددلاین پروژه تون اطلاع خواهد داده شد.
- هر دو عضو گروه باید دفاع کنند و هر دو باید تمام کد و الگوریتم را بلد باشند.
- زبان های مجاز برای کد زدن:

++C ○

C ○

Java ○

Python ○

#C ○

ورودی

در خط اول سایز ماتریس $N * N$ می آید

$$2 \leq N \leq 100$$

در N خط بعد ماتریس ورودی وارد میشود.

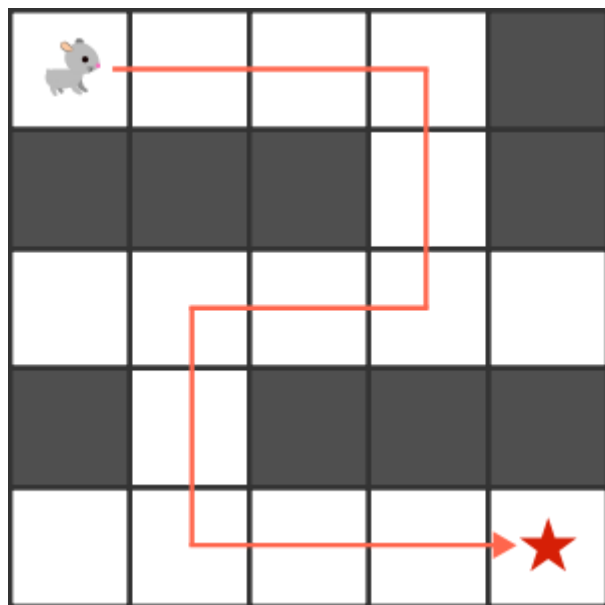
خروجی

در صورتی پروژه تان گرافیکی نیست، مسیر را در ماتریس ورودی با عدد 2 در هر خانه چاپ کنید.

مثال

توجه کنید اگر به صورت گرافیکی پیاده سازی میکنید نیاز به خروجی به صورت مثال ها نیست.

توجه: ورودی هارا باید بتوانید از فایل به صورت زیر بخوانید.



ورودی نمونه ۱

5
 0 0 0 0 1
 1 1 1 0 1
 0 0 0 0 0
 1 0 1 1 1
 0 0 0 0 0

خروجی نمونه ۱

2 2 2 2 1
 1 1 1 2 1
 0 2 2 2 0
 1 2 1 1 1
 0 2 2 2 2

جاهایی که مسیر طی شده 2 بگذارید.

ورودی نمونه ۲

10

```
0 0 0 0 1 0 0 0 1 0
0 1 1 0 1 0 1 0 1 0
0 0 1 0 1 0 1 0 1 0
1 0 1 0 0 0 1 0 0 0
0 0 0 1 1 1 1 1 1 1
0 1 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 1 0 0 0
1 1 1 1 1 0 1 0 1 1
0 0 0 0 0 0 1 0 0 0
```

خروجی نمونه ۲

```
2 0 0 0 1 0 0 0 1 0
2 1 1 0 1 0 1 0 1 0
2 2 1 0 1 0 1 0 1 0
1 2 1 0 0 0 1 0 0 0
0 2 2 1 1 1 1 1 1 1
0 1 2 2 2 2 2 2 2 2
0 1 1 1 1 1 1 1 1 2
0 0 0 0 0 0 1 2 2 2
1 1 1 1 1 0 1 2 1 1
0 0 0 0 0 0 1 2 2 2
```

توجه:

- تقلب و کپی کردن منجر به صفر گرفتن در این پروژه می شود.
- در صورت استفاده از منبع اینترنتی یا کتابی لازم است ذکر شود و رفرنس بهش داده شود
- در صورتی فایل پروژه خود را به فرمت Zip فشرده کردید ولی همچنان بالا 20 مگ بود، کتابخانه ها و فایل های اضافی IDE خود را در مسیر پروژه حذف کنید و فقط فایل های مهم از جمله سورس کد ها رو نگه دارید.
- کوئرا فایل فشرده شده RAR قبول نمیکند، حتما به فرمت ZIP فشرده کنید.

- انجام کارهایی که به عنوان نمره اضافه ذکر شده است نمره چشمگیری برای شما در انتها خواهد داشت، مخصوصا انجام پروژه جوری که ظاهر گرافیکی داشته باشد. شدیداً پیشنهاد میکنم استفاده از کتابخانه های گرافیکی GUI را در زبانی که کار میکنید یاد بگیرید. واقعا ارزشش رو داره، هرچقدر هم ابتدایی باشه ارزش دارد.