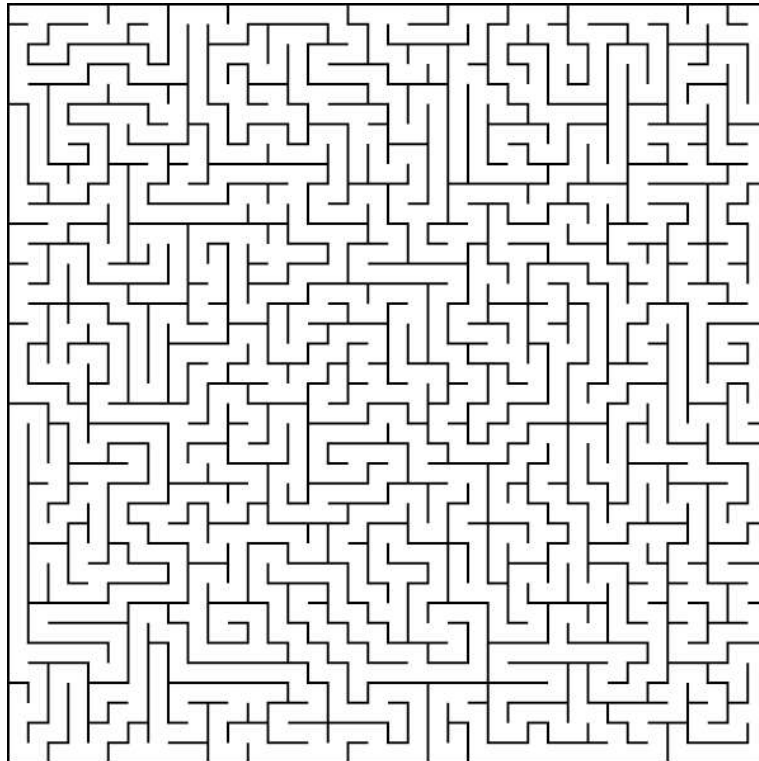




۳ راه نجات



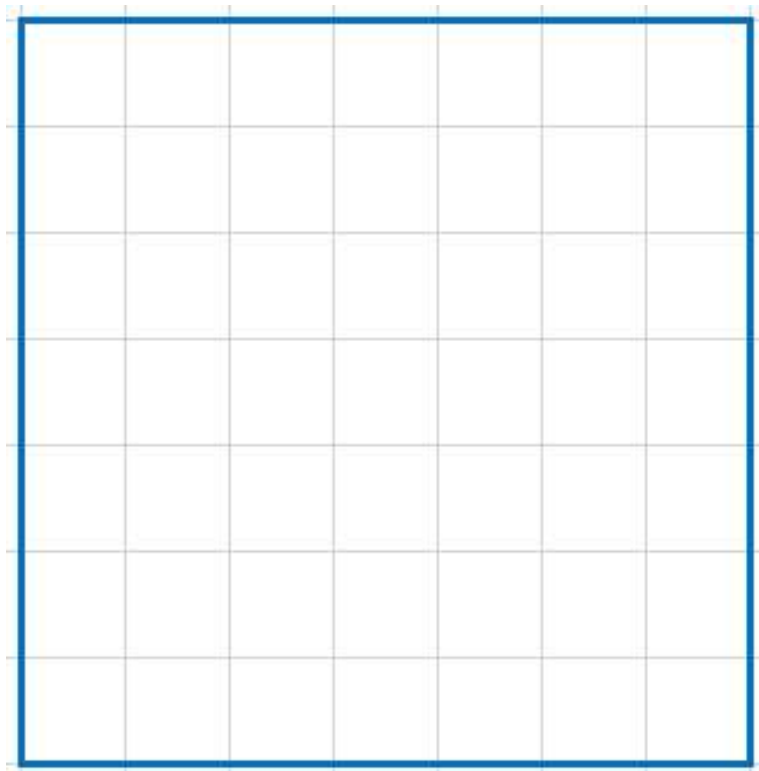
توماس بعد از تلاش های فراوان متوجه می شود که دقیقا در ابتدای یک ماز قرار دارد و تنها راه نجاتش گذر و عبور از این جدول ماز است. پس دست به کار می شود تا راه نجات را پیدا کند. او پس از چندین روز تلاش ناامید می شود اما درمی یابد که در مازی با طول m (تعداد ردیف های جدول ماز) و عرض n (تعداد ستون های جدول ماز) گیر افتاده است. در همین حین یادش می افتد که لپ تاپی را با خود به این سفر عجیب و غریب آورده است تا در مسافرت از دلایل هایش عقب نماند! پس به سرعت سراغ کیفش می رود و با هیجان درمی یابد که لپ تاپ هنوز سالم هست و کار می کند؛ اما لپ تاپ تنها ۱۵ درصد شارژ دارد. پس توماس از شما می خواهد هر چه سریع تر به او کمک کنید تا برنامه ای بنویسد که به او برای یافتن نقشه احتمالی این ماز کمک کند. توماس می داند در هر روز توانایی t بار بررسی کل ماز را دارد پس باید برنامه ای بنویسد که با گرفتن m (طول ماز) و n (عرض ماز)، t حالت و نقشه متفاوت از ماز را ترسیم کند تا او بتواند با آن نقشه ها به بررسی ماز بپردازد. نکته ای که باید به آن دقت کنید این است که توماس اصلا فرصت برای بررسی نقشه های



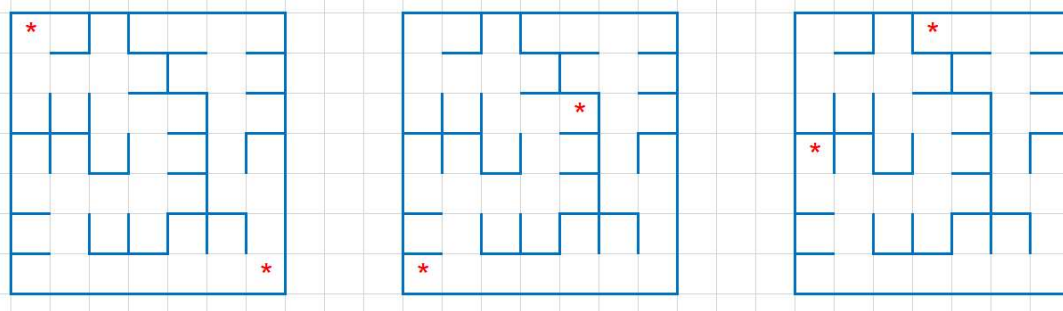
تکراری را ندارد و باید هر t نقشه خروجی برنامه شما حداقل در یک خانه تفاوت داشته باشند!

ماز باید ویژگی‌های زیر را داشته باشد:

- تمامی دیوارهای داخل ماز، حداقل به یک دیوار متصل هستند و دیواری وجود ندارد که به هیچ دیواری متصل نباشد. همچنین تمامی دیوارها به صورت مستقیم یا غیرمستقیم به دیواره اطراف ماز وصل هستند.
- بین هر دو خانه مربعی ماز، تنها یک مسیر یکتا وجود دارد.



دیواره دور ماز



بین هر دو خانه در ماز دقیقا یک مسیر یکتا وجود دارد

این ماز قابل مدل کردن با گراف‌ها است. یکی از الگوریتم‌های پیمایش گراف که در حل این مسئله به شما کمک می‌کند، الگوریتم dfs است که برای فهم آن می‌توانید از این [لینک](#) و یا سایر منابع موجود در اینترنت استفاده کنید.

ورودی

ورودی به ترتیب شامل سه عدد m و n و t است. m عرض ماز و n طول آن است. همچنین t تعداد مازهای متفاوتی است که باید در خروجی چاپ شود.

$$1 \leq n, m, t \leq 100$$

خروجی

خروجی به تعداد t جدول به عرض $1+2*m$ و طول $1+2*n$ خواهد بود که هریک به فرمت زیر است. تمامی خانه‌های ماز (نقاط قرمز رنگ نشان داده شده) می‌بایست با $*$ در خروجی نشان داده شوند. تمام دیواره‌های اطراف هر خانه از جدول ماز بدین صورت نشان داده می‌شوند که در صورت وجود دیوار آن را با 1 و در غیر این صورت (در صورتی که دیواری بین دو خانه وجود نداشت و امکان عبور وجود داشت) آن را با 0 نشان می‌دهیم. همچنین برای ماز یک ورودی و یک خروجی در نظر گرفته شده است که جای ثابتی دارند که در شکل نیز مشخص است. این ورودی و خروجی ماز با کاراکتر e نمایش داده می‌شوند. در تصویر زیر یک ماز $7*7$ و نحوه خروجی دادن آن مشخص شده است:



ورودی نمونه

3 6 3

خروجی نمونه

```
1e11111111111111
1*0*0*0*0*0*1
11101111111101
1*1*0*1*0*1*1
1011111010101
1*0*0*0*1*0*1
11111111111e1
```

```
1e11111111111111
1*1*0*0*0*0*0*1
10111111110101
1*0*0*0*0*1*1
1111111111101
1*0*0*0*0*0*1
11111111111e1
```

```
1e11111111111111
1*0*0*1*0*1*1
1011101010101
1*0*1*0*1*0*1
1110111111101
1*0*1*0*0*0*1
11111111111e1
```

توجه داشته باشید که با توجه به اینکه ماژها به طور رندوم ساخته می‌شوند، این تنها یک خروجی درست برای این ورودی است.

ورودی نمونه

4 11 3

خروجی نمونه

```

1e1111111111111111111111111111l
1*0*1*0*0*1*1*0*0*1*0*1
11101011101010101011101
1*0*1*1*1*1*0*1*1*0*1*1
10111010101111101110101
1*1*0*1*1*0*1*0*1*0*1*1
10111010111010111011101
1*0*0*1*0*0*0*1*0*0*0*1
111111111111111111111111le1
```

[illegible]

```
lelllllllllllllllllll
1*0*1*0*1*0*0*1*0*0*0*1
1010101011101010111101
1*1*0*1*1*0*1*0*1*0*1*1
11111110101111111010101
1*0*1*0*1*1*0*0*0*1*0*1
1010101110101111111111
1*1*0*0*0*1*0*0*0*0*0*1
11111111111111111111le
```