

## درخت بزرگ

- سطح سوال : آسان
- طراح سوال : شایان حقیقت

ابتدا پروژه اولیه را این لینک دانلود کنید.

در دنیای جغدهای گاهول، "درخت بزرگ" خانه و پناهگاه هزاران جغد است. برای حفظ امنیت و رفاه جغدها، این درخت به منابعی مانند غذا، داروهای جادویی و تجهیزات نیاز دارد. جغدها موظف هستند منابع را جمع‌آوری کرده و به انبار این درخت اضافه کنند. تنها **جغدهای مجاز** اجازه دارند این منابع را مدیریت کنند (یعنی می‌توانند منابع را اضافه یا از آن برداشت کنند). جغدهای **غیرمجاز** نمی‌توانند منابعی را اضافه یا برداشت کنند.

### جزئیات پیاده‌سازی:

#### 1. کلاس Owl

این کلاس نماینده جغدهاست.

#### پراپرتی‌ها (Properties):

- . name (String): نام جغد.
- . authorized (boolean): مشخص می‌کند که آیا جغد مجاز است یا خیر.

```
1 | private TODO String name;  
2 | private boolean authorized;
```

#### سازنده (Constructor):

- مقدار name و authorized را دریافت کرده و جغد را ایجاد می‌کند.

#### متدها:

- getName() : نام جغد را برمی‌گرداند.

- `isAuthorized()` : مشخص می‌کند که جغد مجاز است یا خیر.

## 2. کلاس Resource

این کلاس نماینده منابعی است که جغدها مدیریت می‌کنند.

### پراپرتی‌ها (Properties):

name (String): نام منبع.

quantity (int): مقدار منبع.

```
1 | private TODO String name;
2 | private int quantity;
```

### متدها:

- `getName()` : نام منبع را برمی‌گرداند.
- `getQuantity()` : مقدار منبع را برمی‌گرداند.
- `addQuantity(int amount)` : به مقدار منبع اضافه می‌کند.
- `removeQuantity(int amount)` : از مقدار منبع کم می‌کند. اگر مقدار کافی وجود نداشته باشد،

**پیام `Insufficient quantity` چاپ شود و تغییری ایجاد نشود.**

- `toString()` : اطلاعات منبع را در قالب `Resource: [name], Quantity: [quantity]` برمی‌گرداند.

## 3. کلاس `ResourceService` (الگوی Singleton)

این کلاس مسئول مدیریت منابع و جغدهای ثبت‌شده است.

### پراپرتی‌ها (Properties):

- `resources` : لیستی از منابع موجود.
- `registeredOwls` : لیستی از جغدهای ثبت‌شده.
- `resourceService` : نمونه Singleton این کلاس.

```

1 | private final List<Resource> resources = new ArrayList<>();
2 | private final List<Owl> registeredOwls = new ArrayList<>();
3 | private static ResourceService resourceService = null;

```

#### متدها:

- getInstance(): نمونه Singleton کلاس را برمی‌گرداند.
- registerOwl(Owl owl): جغد را ثبت می‌کند. اگر جغد قبلاً ثبت شده باشد، پیام مناسب چاپ می‌شود و مقدار false برمی‌گردد. در غیر این صورت، مقدار true برگردانده می‌شود.
- addResource(Owl owl, Resource resource): منبعی را اضافه می‌کند. اگر جغد مجاز باشد، منبع اضافه شده و مقدار true برمی‌گردد. در غیر این صورت، مقدار false .
- removeResource(Owl owl, String resourceName, int quantity): از منابع موجود کم می‌کند. اگر جغد مجاز نباشد یا منبع کافی نباشد، مقدار false برمی‌گرداند.
- getResources(): لیستی از منابع موجود را برمی‌گرداند.
- clearResources(): تمام منابع را پاک می‌کند.

#### 4. کلاس OwlAgent

این کلاس واسطی برای جغدهای ثبت‌شده و مجاز به سیستم منابع است.

#### ویژگی‌ها (Properties):

- owl: جغدی که از این واسط استفاده می‌کند.

```

1 | private final String name;
2 | private boolean authorized;

```

#### متدها:

- createAgent(Owl owl): واسطی برای جغد ایجاد می‌کند. اگر جغد غیرمجاز باشد یا در سیستم ثبت نشده باشد، مقدار null برمی‌گرداند.
- addResource(String resourceName, int quantity): منبعی را اضافه می‌کند. اگر موفق باشد، مقدار true برمی‌گرداند.

- `removeResource(String resourceName, int quantity)` : منبعی را حذف می‌کند. اگر موفق باشد، مقدار `true` برمی‌گرداند.
- `viewResources()` : منابع موجود را در قالب متنی نمایش می‌دهد.

## قوانین و نکات مهم:

۱. **ثبت جغدها:** جغدها باید در سیستم ثبت شوند تا بتوانند منابع را مدیریت کنند.
۲. **اضافه کردن منابع توسط جغدهای مجاز:** تنها جغدهای مجاز می‌توانند منابع را به انبار اضافه کنند.
۳. **برداشت منابع توسط جغدهای مجاز:** تنها جغدهای مجاز می‌توانند منابع را از انبار برداشت کنند.
۴. **منابع نمی‌توانند منفی شوند:** اگر مقدار مورد نظر برای کم کردن بیشتر از مقدار موجود باشد، عملیات باید لغو شود و پیام مناسب چاپ شود.
۵. **مشاهده وضعیت منابع:** جغدها می‌توانند از طریق متد `viewResources` لیست منابع موجود را مشاهده کنند.
۶. **پیام‌های خطا:** برای هر گونه خطا (مانند مجاز نبودن جغد یا کافی نبودن منابع)، پیام‌های مناسب چاپ شود.

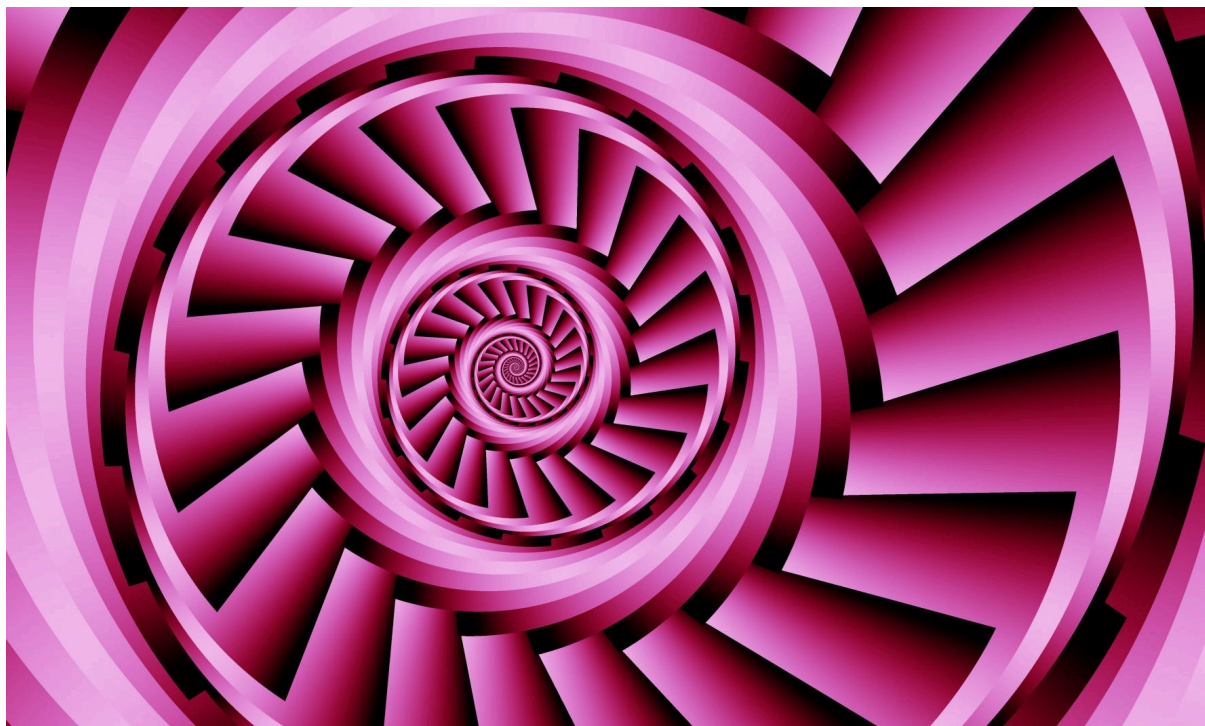
## خروجی فایل:

پروژه خود را در قالب فایل `zip` با ساختار زیر ارسال کنید:

```
<project_name>.zip
├── Owl.java
├── Resource.java
├── OwlAgent.java
└── ResourceService.java
```

## ماتریس مارپیچی

- محدودیت زمان: 4 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: متوسط
- طراح: مهدی کریمی



### ورودی

در خط اول ورودی به شما یکی از اعداد 0 یا 180 داده می شود، که به ترتیب به معنای دوران ماتریس به اندازه 0 و 180 درجه در جهت عقربه های ساعت است. در خط دوم، تعداد سطر و ستون به ترتیب داده می شود. و سپس ماتریس مدنظر به شما داده خواهد شد.

### خروجی

ماتریس داده شده را پس از دوران، به صورت مارپیچی (spiral traversal) پیمایش کنید و ترتیب اعضای آن را چاپ کنید.

## محدودیت ها

$$m = matrix.length$$

$$n = matrix[i].length$$

$$1 \leq m, n \leq 10$$

$$-100 \leq matrix[i][j] \leq 100$$

## ورودی نمونه ۱

```
0
3 3
1 2 3
4 5 6
7 8 9
```

## خروجی نمونه ۱

```
1 2 3 6 9 8 7 4 5
```

▼ توضیح

به دلیل دوران صفر درجه، ماتریس تغییری نمی کند. این ماتریس 3 سطر و 3 ستون دارد و

1 →	2 →	3 ↓
4 →	5	6 ↓
7 ↑	8 ←	9

در نتیجه ترتیب اعضا به صورت 5 4 7 8 9 6 3 2 1 می باشد.

## ورودی نمونه ۲

180  
3 3  
1 2 3  
4 5 6  
7 8 9

## خروجی نمونه ۲

9 8 7 4 1 2 3 6 5

▼ توضیح

پس از دوران 180 درجه، ماتریس به شکل زیر در می آید. این ماتریس 3 سطر و 3 ستون دارد و

9	→	8	→	7
6	→	5		↓
				↓
↑				↓
3	←	2	←	1

در نتیجه ترتیب اعضا به صورت 5 6 3 2 1 4 7 8 9 می باشد.

## در جستجوی کلمه

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: سخت
- طراح: آریا زریاب

راب و باب دو دوست قدیمی هستند که علاقه زیادی به بازی های چالشی دارند. این بار نوبت باب است که با گرفتن امتیازِ نوبتش از راب جلو بیفتد ، او به راب یک جدول با ابعاد  $m$  در  $n$  میدهد که در هر خانه آن یک کرکتر وجود دارد. سپس چند کلمه انتخاب میکند که راب باید آن ها را از جدولش پیدا کند.

راب که دوست ندارد بازی را ببازد از شما کمک میخواهد تا جواب را بدون اینکه باب بفهمد تقلب کرده به او بگویید..

هر کلمه میتواند از حروف خانه های متوالی مجاورش ساخته شود که خانه های مجاور به صورت افقی یا عمودی همسایه هستند. همچنین یک خانه (یک حرف) را نمیتوان بیش از یکبار در هر مسیر استفاده کرد. برای بهتر متوجه شدن خواسته مسئله به مثال ها نگاه کنید.

## ورودی

در خط اول ورودی دو عدد  $m$  و  $n$  به ترتیب آمده است که ابعاد جدول است.

سپس در  $m$  سطر بعدی در هر سطر  $n$  کرکتر آمده است که نشان دهنده حرف در آن مکان جدول است.

سپس در سطر بعدی عدد  $t$  آمده است که تعداد کلماتی است که باب از راب میخواهد.

در  $t$  سطر بعدی در هر سطر یک کلمه آمده است.

$$1 \leq n, m \leq 10$$

$$1 \leq t \leq 5$$



$$1 \leq wordLength \leq 8$$

تضمین میشود که تمام حروف جدول ، حروف بزرگ انگلیسی هستند.

## خروجی

در t خط خروجی در هر خط پاسخ مسئله برای آن کلمه است.

اگر کلمه پیدا شود YES در غیر این صورت NO چاپ شود.

## مثال

### ورودی نمونه ۱

3 4  
ABCE  
SFCS  
ADEE  
1  
ABCCED

### خروجی نمونه ۱

YES

توضیحات ▼

A	B	C	E
S	F	C	S
A	D	E	E

همانطور که در تصویر مشاهده میکنید با انتخاب مسیر نارنجی به کلمه میرسیم.

ورودی نمونه ۲

4 4  
ARXC  
PIAB  
AIRO  
BDAB  
4  
ROB  
ARIA  
BOB  
XARB

خروجی نمونه ۲

YES  
YES  
YES  
NO

توضیحات ▼

A	R	X	C
P	I	A	B
A	I	R	O
B	D	A	B

همانطور که در تصویر مشاهده میکنید با انتخاب مسیر زرد یا آبی به ARIA

با انتخاب بنفش به BOB

با انتخاب سبز به ROB میرسیم.

ولی هیچ کلمه ی XARB ای پیدا نمیشود

▼ راهنمایی

سعی کنید یک رابطه بازگشتی برای پیمایش جدول پیدا کنید.

## در جستجوی الی (امتیازی)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: سخت
- طراح: آرمان اسدی

بعد از اینکه جول با درخواست مارلین برای تحویل الی به گروه فایرفلایز که در ساختمان کاپیتول مستقر بودند موافقت کرد، گروهی از کلیکرها در میانه راه به آنها حمله می‌کنند. جول و الی برای فرار کردن از دست آنها وارد هزارتویی می‌شوند و در آنجا یکدیگر را گم می‌کنند.



این هزارتو از بالا به شکل یک مش  $n$  در  $n$  است. الی در نقطه ای از این مش گیر افتاده است و جول باید او را پیدا کند. ساختار هزارتو به گونه ای است که جول در آن فقط می‌تواند به سمت **راست** یا **پایین** حرکت کند. مشکل اصلی ای که جول با آن مواجه است، وجود کلیکرها در نقاطی از این مش است و متأسفانه مهمات جول تمام شده است و او **نمی‌تواند** از نقاطی که کلیکرها در آن وجود دارند، عبور کند. شما باید به جول کمک کنید تا الی را پیدا کند و او را نجات دهد.

## ورودی

در اولین خط ورودی عدد  $n$  به شما داده می‌شود که بیانگر ابعاد مش است.

$$1 \leq n \leq 30$$

سپس در  $n$  خط بعدی در هر خط  $n$  کاراکتر به شما داده می‌شود که نشان دهنده وضعیت مش است. به شکل زیر دقت کنید:

```
.J..X
X....
.X..X
.....
..X.E
```

- J : نشان دهنده موقعیت جول است.
- E : نشان دهنده موقعیت الی است.
- X : نشان دهنده موقعیت هر کلیکر است.
- . : نشان دهنده خالی بودن فضا است.

**دقت کنید که جول فقط می‌تواند در فضای خالی حرکت کند.**

## خروجی

خروجی شامل دو خط است:

۱. در خط اول خروجی تعداد حالاتی را که جول می‌تواند به الی برسد را چاپ کنید. **تضمین می‌شود که**

**همواره مسیری از جول به الی وجود داشته باشد.**

۲. در خط دوم خروجی یکی از مسیرهایی را که جول می‌تواند به الی برسد را چاپ کنید. به عنوان مثال

یکی از مسیرهایی که برای مثال بالا وجود دارد به شکل زیر است:

```
RDDDRDR
```

- R : حرکت به سمت راست.
- D : حرکت به سمت پایین.

توجه داشته باشید که در مسیر خروجی ترتیب حرکات جول از چپ به راست در نظر گرفته میشود.

## مثال

برای فهم بهتر سوال به مثال های زیر دقت کنید.

### ورودی نمونه ۱

3  
J.X  
...  
X.E

### خروجی نمونه ۱

4  
RDRD

▼ توضیح مثال

در این مثال به ۴ طریق جول می تواند به الی برسد:

۱. RDRD
۲. RDDR
۳. DRRD
۴. DRDR

### ورودی نمونه ۲

4  
.J..  
X..X  
X.X.  
...E

خروجی نمونه ۲

1  
DDRR

ورودی نمونه ۳

5  
.J..X  
X....  
.X..X  
.....  
..X.E

خروجی نمونه ۳

16  
RDDDRDR

## مستطیل ویژگی ها

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح : متوسط
- طراح: امیرعلی وکیلی

در این مسئله به شما جدولی از اعداد مثل  $a_{i,j}$  داده شده است. شما باید بین همه‌ی زیر مستطیل‌ها زیر مستطیلی را بیابید که جمع اعداد داخلش بیشینه باشد.

## ورودی

در خط اول به شما  $n$  و  $m$  داده می‌شود که اولی تعداد ردیف‌ها و دومی تعداد ستون‌های جدول است.

در  $n$  خط بعدی نیز در هر خط به شما  $m$  عدد داده می‌شود. عدد  $j$ ام در خط  $i$ ام برابر  $a_{i,j}$  است.

$$1 \leq n, m \leq 500$$

$$-10^9 \leq a_i \leq 10^9$$

## خروجی

تنها یک عدد برابر بیشینه جمع عناصر بین همه‌ی زیر مستطیل‌های جدول را خروجی دهید. زیر جدول تهی نداریم!

## مثال

### ورودی نمونه

```
3 3
-3 4 1
```



5 7 -3  
-10 -11 -100

خروجی نمونه

13

چهار عنصر بالا-چپ یک جواب برای مسئله است. ( $-۳ + ۴ + ۵ + ۷ = ۱۳$ )

