

## مساحت محصور

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

تعدادی تخته با شماره‌های ۱ تا  $n$  در کنار هم داده شده است. ارتفاع تخته‌ی  $i$ ام  $h_i$  متر و عرض آن ۱ متر است. می‌خواهیم مستطیل با بیشترین مساحت محصور بین این  $n$  تخته را بیابیم. منظور از مستطیل محصور بین تخته‌ها، مستطیلی است که سطح آن تماماً درون تخته‌ها قرار گیرد.

## ورودی

در خط اول ورودی عدد  $n$  و در خط بعد  $n$  عدد صحیح نامنفی داده می‌شود که عدد  $i$ ام نشان‌دهنده‌ی ارتفاع تخته‌ی  $i$ ام است.

$$1 \leq n \leq 10^6$$

## خروجی

در تنها خط خروجی باید مساحت مستطیل خواسته شده را چاپ کنید.

## مثال

### ورودی نمونه

```
6
2 7 5 6 3 1
```

### خروجی نمونه

15

## فرزاد کارکن

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

حالا که امتحان های میان ترم فرزاد تمام شده است و زمان بیشتری دارد، او به فکر کار افتاده است. پس از جستجوهای فراوان نهایتاً در شرکت دانیال اینا کاری به او داده شد. کار او به این صورت است که به او چند عدد صحیح می دهند که میزان سود یا ضرر شرکت در روزهای متوالی است. (واحد اعداد میلیون تومان است.) او باید بگوید بیشترین سود شرکت چقدر بوده است. مثلاً در روز اول به او این عددها را دادند: 1, 2, -5, 4, -3, 2.

واضح است که بیشترین سود شرکت در چهارمین روز بوده است، که برابر ۴ میلیون تومان است. چون مجموع اعضای هر زیر آرایه دیگری از این آرایه داده شده، مقداری کوچک تر از ۴ دارد. دقت کنید که اگر همه اعداد، منفی (ضرر) بودند، میزان سود برابر ۰ است. برنامه‌ای بنویسید که فرزاد به وسیله آن بدون محاسبات ذهنی، کار خود را انجام دهد.

## ورودی

در خط اول ورودی تعداد روزهایی که قرار است سود و ضرر و در ادامه آرایه‌ی سود و ضررها در این روزها گرفته می‌شود.

$$1 \leq n \leq 100$$

## خروجی

در خروجی شما باید میزان بیشترین سود را بیان کنید. به ورودی و خروجی نمونه دقت کنید.

## مثال

### ورودی نمونه ۱

12

7 -1 -2 1 5 -11 9 1 4 -1 3 -10

خروجی نمونه ۱

16

توضیح خروجی: بیشترین سود شرکت در روزهای ۷ تا ۱۱ است که مجموع اعداد شماره ۷ تا ۱۱ برابر ۱۶ است.

ورودی نمونه ۲

5

-5 -2 -9 -1 -3

خروجی نمونه ۲

0

## دترمینان ماتریس‌ها

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

از آنجایی که دترمینان یک ماتریس بسیار مفید و کاربردیست!

برنامه‌ای بنویسید که ابتدا  $n$  و سپس درایه‌های یک ماتریس  $n \times n$  را بگیرد. و با کمک تابع بازگشتی دترمینان ماتریس را محاسبه و با دقت دو رقم اعشار چاپ کند.

### ورودی

در خط اول ورودی عدد  $n$  آمده است. در  $n$  خط بعد در هر خط  $n$  عدد گویا آمده که درایه‌های ماتریس را مشخص می‌کنند. (هر درایه‌ی ماتریس عددی گویاست که قدرمطلق آن از ۱۰۰ کمتر است.)

$$1 \leq n \leq 10$$

### خروجی

در خروجی دترمینان ماتریس داده شده را تا ۲ رقم اعشار چاپ کنید.

### مثال

#### ورودی نمونه ۱

```
3
1.0 0.0 0.0
2.0 3.0 4.0
5.0 6.0 7.0
```

خروجی نمونه ۱

-3.00

ورودی نمونه ۲

2

1.1 2.2

3.3 4.4

خروجی نمونه ۲

-2.42

## ماتریس دختر خوشگل

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه ای بنویسید که ابتدا ماتریس  $m$  با 7 سطر و ستون از جنس صحیح تعریف کند. سپس داده های ماتریس را از ورودی خوانده و بصورت سطری در آن بریزد. در گام بعد عناصر مثلث بالا را با مثلث پایین جابجا کند. همچنین مثلث راست را با مثلث چپ جابجا نماید. در آخر اعداد ماتریس را چاپ نماید.

Green	Green	Green	Green	Green	Green	Green
Gray	Green	Green	Green	Green	Green	Gray
Gray	Gray	Green	Green	Green	Gray	Gray
Gray	Gray	Gray	Green	Gray	Gray	Gray
Gray	Gray	Gray	Gray	Gray	Gray	Gray
Gray	Gray	Gray	Gray	Gray	Gray	Gray
Gray	Gray	Gray	Gray	Gray	Gray	Gray

برای مثال در شکل بالا مثلث سبز با مثلث آبی و مثلث قهوه ای با مثلث خاکستری جابجا میشود.

## ورودی

ورودی تنها شامل یک ماتریس  $7 \times 7$  میباشد.

## خروجی

خروجی ماتریس حاصل از جواب مساله میباشد.

## مثال

### ورودی نمونه ۱

```
1 2 3 4 5 6 7
2 3 4 5 6 7 8
3 4 5 6 7 8 9
4 5 6 7 8 9 10
5 6 7 8 9 10 11
6 7 8 9 10 11 12
7 8 9 10 11 12 13
```

### خروجی نمونه ۱

```
1 8 9 10 11 12 7
8 3 8 9 10 7 2
9 8 5 8 7 4 3
10 9 8 7 6 5 4
11 10 7 6 9 6 5
12 7 4 5 6 11 6
7 2 3 4 5 6 13
```



## کان چیچی شن؟ /:

### کانولوشن Convolution

**آرایه‌های درست حسابی!** در این تمرین می‌خوایم ببینیم که کاربرد واقعی آرایه‌ها و اینا کیلویی چند؟ و ببینیم که کاربرد واقعیش کجاست و با یه سری مفهوم‌های آشناشیم که خیلی جذاب و جالب هستن! بدین شکل <= 😊

😊 آرایه‌ها رو که خوندید پس توضیحی نمیدیم (دوست داریم ^\_^) بریم سراغ یه مفهوم و موضوع جدیدی که خیلی هم خوبه و سلام می‌رسونه به اسم پردازش تصویر (=) پردازش تصویر در خیلی چیزها کاربرد داره. همین افکت‌هایی که تو اسنپ‌چت تر و تر استفاده میکنین و می‌خندین دور هم و ... اینا همه پردازش تصویره یا مثلاً یه جاهایی هم جون ملت رو نجات میده مثلاً لخته شدن خون تو رگ‌های مغزی رو تشخیص میده.

می‌خوایم ببینیم که چطوری دقیقاً لبه تصاویر رو تشخیص میدن تو تصویر؟ اگه دقت کرده باشین مثلاً این افکت‌هایی که صورتتون رو می‌ذاره رو یه چیز دیگه خب صورتتون رو از کجا می‌فهمه؟ تاحالا بهش فکر کردین؟

جواب اینه که میاد حدود صورتتون رو تشخیص میده که کجاست و بعد میاد بر اون اساس افکت رو روی صورتتون اعمال میکنه.

حالا می‌خوایم در این تمرین برای اینکه با آرایه‌ها بیشتر کار کنیم و اینا بیایم ببینیم که چطوری میشه یه لبه رو توی تصاویر تشخیص داد؟

اگه توی نت سرچ کنید و مثلاً این لینک ([https://en.wikipedia.org/wiki/Sobel\\_operator](https://en.wikipedia.org/wiki/Sobel_operator)) رو ببینین

می‌فهمین که یه ماتریسی هست که اگه در ماتریس عکسمون ضرب بشه حاصل یه ماتریسی میشه که توش میشه دید که لبه داریم یا نه.

## Vertical edge detection examples

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

\*

1	0	-1
1	0	-1
1	0	-1

=

0	10	10	0
0	10	10	0
0	10	10	0
0	10	10	0



به این عملیات کانولوشن میگن .

حالا بریم ببینم که چطوری این کار میکنه.

به این ماتریس میگن sobel matrix که تو شکل پایین میبینید. این ماتریس یه ماتریس ۳ در ۳ هست که برای تشخیص لبه های عمودی یا افقی تصاویر هست.

## Vertical and Horizontal Edge Detection

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

کانولوشن یه کم ضربش متفاوته. همونطور که تو شکل پایین میبینید :

## Vertical edge detection

$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

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

6x6

"convolution"

$$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \times \begin{matrix} 3 & 0 & 1 \\ 1 & 5 & 8 \\ 2 & 7 & 2 \end{matrix}$$

3x3 filter

=

-5			

4x4

Andrew Ng

طریقه ضرب کردنش اینطوریه که شما باید یک پنجره ۳ در ۳ (همین قسمتی که آبی رنگ شده) رو در نظر بگیرید. این ماتریس رو درایه به درایه در فیلتر sobel که میبینید ضرب بکنید. ضرب های انجام شده رو با رنگ سبز اون بالا نوشته. تهش به شما ی عدد میده که میشه درایه اول این ماتریس خروجی.

حالا مرحله بعد اینه که این پنجره رو به اندازه یک خونه

شیفت بدید. شیفت دادن رو در تصویر پایین میتونین ببینید که اینجا به اندازه یک خونه شیفت میده به راست تا اینکه به انتها برسه بعد برمیگرده اول و یک خط میاد پایین و همینطور تا آخر تو هر مرحله دوباره کل اون ناحیه آبی رنگ جدید که میشه پنجره جدیدتون رو در اون فیلتر ضرب می کنید تا اینکه ماتریس خروجی تمام عددهاش بدست بیان.

3	0 <sup>1</sup>	1 <sup>0</sup>	2 <sup>1</sup>	7	4	3	0	1 <sup>1</sup>	2 <sup>0</sup>	7 <sup>1</sup>	4	3	0	1	2 <sup>1</sup>	7 <sup>0</sup>	4 <sup>1</sup>	3	0	1	2 <sup>1</sup>	7 <sup>0</sup>	4 <sup>1</sup>
1	5 <sup>1</sup>	8 <sup>0</sup>	9 <sup>1</sup>	3	1	1	5	8 <sup>1</sup>	9 <sup>0</sup>	3 <sup>1</sup>	1	1	5	8	9 <sup>1</sup>	3 <sup>0</sup>	1 <sup>1</sup>	1 <sup>1</sup>	5 <sup>0</sup>	8 <sup>1</sup>	9 <sup>0</sup>	3 <sup>1</sup>	1 <sup>1</sup>
2	7 <sup>1</sup>	2 <sup>0</sup>	5 <sup>1</sup>	1	3	2	7	2 <sup>1</sup>	5 <sup>0</sup>	1 <sup>1</sup>	3	2	7	2	5 <sup>1</sup>	1 <sup>0</sup>	3 <sup>1</sup>	2 <sup>1</sup>	7 <sup>0</sup>	2 <sup>1</sup>	5 <sup>0</sup>	1 <sup>1</sup>	3 <sup>1</sup>
0	1	3	1	7	8	0	1	3	1	7	8	0	1	3	1	7	8	0 <sup>1</sup>	1 <sup>0</sup>	3 <sup>1</sup>	1	7	8
4	2	1	6	2	8	4	2	1	6	2	8	4	2	1	6	2	8	4	2	1	6	2	8
2	4	5	2	3	9	2	4	5	2	3	9	2	4	5	2	3	9	2	4	5	2	3	9
6×6						6×6						6×6						6×6					

حالا این یعنی چی؟

## Vertical edge detection examples

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 10 & 10 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & 10 & 10 & 0 \end{bmatrix}$$

اگه تو این تصویر نگاه کنی ماتریسی که به شما داده شده یک طرف رنگ تیره هست و یک طرف رنگ روشن. یعنی لبه داریم. حالا تو خروجی رو اگه نگاه بکنی جایی که دقیقا لبه هست رو مشخص کرده یعنی دو ردیف وسط دارای مقدار هست و باقی مقداری ندارن.

به همین سادگی!

حالا کاری که باید شما تو این تمرین بکنین اینه که شما باید یک ماتریس 6 در 6 از ورودی بگیرید(مثلا میتونید همون مثال بالا رو هم امتحان کنید) و بعد بیاید همون فیلتر عمودی که بالا توضیح داده شده رو روش اعمال کنید و ماتریس نهایی رو برای ما چاپ کنید. کد این سوال توسط کوئرا تصحیح نمیشه و ما دستی کداتونو چک میکنیم!

- تمرین اصلا چیز سختی نیست. صرفا خواستیم که یه کمی کاربردی باشه و بعضی چیزارو یاد بگیرید. یک پیاده سازی خیلی کوچیکه.

- اگه سوالی هست همین جا مطرح کنین که جوابتونو بدیم.

- تمام عکس‌هارو از کورس آقا Andrew Ng برداشتم.

## ماتریس پسر خوشگل (اختیاری)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه ای بنویسید که ابتدا ماتریس  $m$  با 7 سطر و ستون از جنس اعشار تعریف کند. سپس داده های ماتریس را از ورودی خوانده و به صورت ستونی در آن بریزد. در گام بعدی عناصر روی قطر اصلی و قطر فرعی را نظیر به نظیر با یکدیگر جمع کرده و در ستون وسط قرار ستون وسط قرار دهد و مابقی داده های ماتریس بدون تغییر می ماند. در آخر اعداد ماتریس را بصورت سطری چاپ نماید.

۶/۱۰			۶۴/۲۵			۵۸/۱۵
	۵/۳		۱۳/۹		۸/۶	
		۷/۴۶	۱۶/۰	۸/۵۴		
			۴/۵+۴/۵ =۹/۰			
		۷/۹	۱۶/۵	۸/۶		
	۹/۶		۵۵/۰		۴۵/۴	
۳۳/۳						

۱۱/۱			۳۴/۳			۱۱/۱
------	--	--	------	--	--	------

## ورودی

در ورودی یک ماتریس  $n \times n$  می آید که به صورت ستونی پر میشود.

## خروجی

در خروجی یک ماتریس  $7 \times 7$  است که جواب مساله ی ما میباشد. بین هر خانه یک اسپیس وجود دارد. و اعداد تا دو رقم اعشار میباشد.

## مثال

### ورودی نمونه ۱

```

1.2 3.4 5.6 7.8 9.10 10.11 11.12
2.3 4.5 6.7 8.9 10.11 12.13 14.15
3.4 5.6 7.8 9.10 1.12 13.14 15.16
4.5 6.7 8.9 10.11 12.13 14.15 16.17
5.6 7.8 9.10 11.12 13.14 15.16 17.18
6.7 8.9 10.11 12.13 14.15 16.17 18.19
7.8 9.10 11.12 13.14 15.16 17.18 19.20

```

### خروجی نمونه ۱

```

1.20 2.30 3.40 9.00 5.60 6.70 7.80
3.40 4.50 5.60 13.40 7.80 8.90 9.10
5.60 6.70 7.80 16.90 9.10 10.11 11.12
7.80 8.90 9.10 20.22 11.12 12.13 13.14
9.10 10.11 1.12 14.26 13.14 14.15 15.16
10.11 12.13 13.14 28.30 15.16 16.17 17.18
11.12 14.15 15.16 30.32 17.18 18.19 19.20

```

## زینی(اختیاری)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک جدول  $n \times m$  داریم که در هر خانه‌اش عددی نوشته شده است.

یک برنامه نویس معمولی به یک خانه از جدول زینی می‌گوید اگر بتوان روی آن نشست! اما یک برنامه نویس نیمبو به یک خانه از جدول زینی می‌گوید اگر ۴ همسایه مجاور ضلعی‌اش موجود باشند و عددش از اعداد خانه های مجاور چپ و راستش بزرگ‌تر، و از اعداد خانه های مجاور بالا و پایینش کوچک‌تر باشد، و یا بالعکس (یعنی عددش از اعداد خانه‌های مجاور چپ و راستش کوچک‌تر و از اعداد خانه‌های مجاور بالا و پایینش بزرگ‌تر باشد).

شما به عنوان برنامه نویسی نیمبو باید تعداد خانه‌های زینی یک جدول را پیدا کنید.

## ورودی

خط اول ورودی شامل دو عدد  $n$  و  $m$  است.

در  $n$  خط بعدی برنامه، سطرهای جدول آمده اند. به طوری که هر خط شامل  $m$  عدد است که نشان‌دهنده اعداد یک سطر از جدول هستند. اعداد جدول طبیعی و کوچکتر مساوی  $10^9$  اند.

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

## خروجی

خروجی شامل یک عدد است که تعداد خانه‌های زینی جدول از دیدگاه برنامه‌نویسی نیمبو را نشان می‌دهد.

## مثال

### ورودی نمونه ۱

3 3  
1 2 3  
6 5 6  
1 1 1

## خروجی نمونه ۱

1

فقط خانه وسط جدول زینی است. دقت کنید که بقیه خانه‌ها هیچ‌کدام شرط داشتن ۴ همسایه را ندارند.

## ورودی نمونه ۲

4 4  
1 2 4 1  
7 4 1 1  
1 3 2 4  
1 4 1 1

## خروجی نمونه ۲

2

خانه‌ای که در سطر سوم و ستون دوم قرار دارد، و همچنین خانه‌ای که در سطر سوم و ستون سوم قرار دارد زینی‌اند.



## ضرب ماتریس‌ها (اختیاری)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

این برنامه ۳ عدد ورودی می‌گیرد که عددهای اول و دوم به ترتیب تعداد سطر و ستون ماتریس اول هستند و عددهای دوم و سوم به ترتیب تعداد سطر و ستون ماتریس دوم هستند؛ سپس مقدار هر درایه ماتریس را گرفته و ضرب دو ماتریس را چاپ می‌کند.

### ورودی

در خط اول ورودی ۳ عدد آمده که نشانگر تعداد سطر و ستون های ۲ ماتریس است.

در ادامه ورودی درایه‌های ماتریس اول و سپس درایه‌های ماتریس دوم آمده است.

تمامی اعداد داده شده در ورودی کوچک‌تر از ۱۰۰ می‌باشند.

### خروجی

در خروجی درایه‌های ماتریس حاصل از ضرب ۲ ماتریس داده شده در ورودی را چاپ کنید.

### مثال

#### ورودی نمونه

```
2 3 2
1 2 3
4 5 6
1 2
3 4
5 6
```

## خروجی نمونه

22 28

49 64