

سازماندهی هوشمند

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



گرو دوست دارد تمام مینیون‌های زیر دست خود را سازماندهی کند. به همین منظور نیاز دارد که از مدل‌های پیشرفته‌ی هوش مصنوعی کمک بگیرد. او می‌داند برای ساخت این مدل‌ها به چند عملیات ریاضی روی ماتریس‌ها نیاز دارد ولی پیاده‌سازی این عملیات را بلد نیست. چون حال و حوصله یادگیری هم ندارد (بالاخره سنی از او گذشته!) پس به شما می‌گوید که برنامه‌ای طراحی کنید تا بتواند این عملیات ساده را انجام دهد، بالاخره مینیون‌ها را سازماندهی کند و نقشه‌های خود را پیش ببرد.

▼ ماتریس‌های چند بعدی (تنسورها)

ماتریس‌های چند بعدی، نسخه‌های تعمیم‌یافته‌ی ماتریس‌های معمولی هستند که می‌توانند بیشتر از دو بُعد داشته باشند.

- ماتریس ۱ بعدی: یک بردار است. مثل یک ردیف یا ستون از اعداد. به عنوان مثال: $[1, 2, 3]$ یک بردار ردیفی (ماتریس 1×3) است.

- ماتریس ۲ بعدی: همان ماتریس معمولی است که از ردیف‌ها و ستون‌ها تشکیل شده است. برای مثال: $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ یک ماتریس 2×2 است.
- ماتریس ۳ بعدی: یک "مکعب" از اعداد که شامل چندین ماتریس ۲ بعدی به صورت پشته‌ای است. به عنوان مثال، می‌توان چندین ماتریس را روی هم قرار داد تا یک ماتریس ۳ بعدی بسازیم. ماتریس‌های چند بعدی می‌توانند به ابعاد بالاتر هم تعمیم داده شوند، که به آن‌ها **تَنسور** می‌گویند.

عملیات:

- عملیات خطی: روی درایه‌های متناظر از ماتریس‌ها، عملیات خطی انجام می‌شود.

$$C = \alpha A + \beta B$$
در ورودی به صورت:

$$L \{ \text{who} \} \alpha \beta$$
عمل بالا به معنی این است که حاصل عملیات در ماتریس `who` ریخته شود. اگر `who` ۱ بود، نتیجه در ماتریس اول و اگر ۲ بود نتیجه در ماتریس دوم ریخته شود. توجه کنید که **ضرایب اعداد صحیح هستند**.
- ترانپازه (Transpose): دو بعد اول از ماتریس‌های ما را عوض می‌کند.

در ورودی به صورت:

T

که یعنی جای بعد ۱ و ۲ ماتریس‌ها عوض می‌شود. اگر ماتریس یک بعدی بود، این عمل تغییری در آن ایجاد نمی‌کند!

توضیح بیشتر برای دستور T: در این سوال `transpose` یک نوع خاص از دستور `reshape` است که جای بعدهای اول و دوم را عوض می‌کند و ترتیب درایه‌های ماتریس را به هم نمی‌زند. به طور مثال:

1 2
3 4

تبدیل به:

1 2
3 4

1 2 3
4 5 6

تبدیل به:

1 2
3 4
5 6

می‌شود.

- تغییر شکل: بدون از دست رفتن اطلاعات به ترتیب اطلاعات ماتریس‌ها را در ماتریس‌های جدید می‌ریزیم.

در ورودی به صورت:

$R \ n \ d1 \ d2 \ \dots \ dn$

یعنی هر دو ماتریس تبدیل به ماتریس n بعدی می‌شوند که $d1 \times d2 \times \dots \times dn$ هستند. تضمین می‌شود که همه‌ی اطلاعات در این ماتریس جا می‌شود و ماتریس خالی هم نمی‌ماند.

شما نباید از آرایه‌ها استفاده کنید و نباید ماتریس ورودی را به صورت یک خط از اطلاعات ذخیره کنید؛ بلکه باید اشاره‌گر (void*) استفاده کنید!

ورودی

- بعد ماتریس‌ها در خط اول داده می‌شود.
- در خط بعدی اندازه‌ی ماتریس‌ها در هر بعد به ترتیب به شما داده می‌شود.
- در خط‌های بعدی دو ماتریس چند بعدی به شما داده می‌شود. درایه‌ها به ترتیب روی یک خط با فاصله می‌آیند. هر درایه‌ی ماتریس بین 100 و 0 است. درایه‌های هر ماتریس از جنس عدد صحیح هستند. اگر درایه‌ای در طول عملیات از این بازه خارج شد، مد 100 آن را به جای خودش قرار دهید.
- در خط بعدی تعداد عملیات و در خط‌های بعدی، هر عمل داده می‌شود.

خروجی

بعد از انجام همه عملیات، دو ماتریس حاصل را در خطهای مجزا با آدرس هر درایه مانند مثالهای داده شده چاپ کنید.

مثالها

ورودی نمونه ۱

```
3
1 2 3
55 38 32 45 73 91
33 70 35 22 49 50
8
T
L 2 21274 6463
T
L 2 5049 19911
L 1 12686 14316
L 1 30700 16741
T
L 2 15132 4230
```

خروجی نمونه ۱

```
Matrix 1:
(1,1,1): 94
(1,1,2): 64
(1,1,3): 11
(2,1,1): 21
(2,1,2): 96
(2,1,3): 3
Matrix 2:
(1,1,1): 28
(1,1,2): 68
```

$(1,1,3): 82$ $(2,1,1): 2$ $(2,1,2): 52$ $(2,1,3): 86$

ورودی نمونه ۲

1

2

29 72

44 80

5

T

R 2 2 1

L 2 17779 12502

T

L 1 28424 13380

خروجی نمونه ۲

Matrix 1:

 $(1,1): 16$ $(1,2): 68$

Matrix 2:

 $(1,1): 79$ $(1,2): 48$