

## اول و آخر

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

یکی از مهم‌ترین بحث‌ها بین عارف و مهرشاد، اهمیت حروف ابتدایی و یا انتهای یک کلمه برای تشخیص آن کلمه است. عارف بر این باور است که حروف ابتدایی بیشترین کمک برای تشخیص یک کلمه را به ما می‌کنند اما مهرشاد برخلاف عارف، بر این باوره که حروف انتهای این قدرت را دارند. آن‌ها از شما می‌خواهند تا با نوشتن یک برنامه به آن‌ها کمک کنید تا کمترین حروف مورد نیاز برای متمایز کردن این کلمات را، هم از اول و هم از آخر مشخص کنید.

شما باید برنامه‌ای بنویسید که با گرفتن یک خط ورودی، کلمات آن را از هم جدا کرده و سپس کمترین تعداد حروف ابتدای هر کلمه که برای تمایز آن از بین کل کلمات لازم است را مشخص کند. همین کار برای حروف انتهای کلمات هم باید انجام شود و به ازای هر کلمه مشخص شود که کمترین تعداد حروف لازم از انتهای کلمه برای تفکیک یک آن کلمه از بقیه کلمات چه تعدادی است.

## ورودی

ورودی تنها شامل یک خط است که در آن یک رشته از کلمات به شما داده می‌شود. تضمین می‌شود که طول این خط حداکثر 200000 کاراکتر است و بین کلمات مختلف "-" وجود دارد اما ممکن است تعدادی space اضافه بین "-" و هر کلمه وجود داشته باشد که نباید در نظر گرفته شوند.

## خروجی

خروجی برنامه‌ی شما باید شامل ۲ خط باشد که در خط اول تعداد حروف از ابتدای کلمات به همان ترتیبی که در رشته ورودی آمده‌اند مشخص شود و در خط دوم تعداد حروف از انتهای کلمات مشخص می‌شود.

## مثال

## ورودی نمونه ۱

duck -clock - clothes

## خروجی نمونه ۱

1 4 4  
3 3 1

همانطور که می بینید duck صرفاً با یک حرف از بقیه کلمات متمایز می شود اما برای clock ما نیاز به cloc و برای clothes نیاز به clot داریم. در نتیجه خروجی خط اول 1 4 4 می باشد. همچنین با توجه به انتهای کلمات، clothes با s خود متمایز می شود اما duck به uck و clock به ock نیازمند است.

## ورودی نمونه ۲

hi

## خروجی نمونه ۲

1  
1

## ورودی نمونه ۳

brick-block-bank-crane-broke-banker

## خروجی نمونه ۳

3 2 4 1 3 5  
3 3 2 2 2 1



## تقسیم اراضی

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

نیما جدیداً مسئولیت تقسیم زمین در یک شهر را به عهده گرفته‌است. قبل از این که نیما بخواهد کار خود را شروع کند، می‌خواهد ببیند آیا هیچ راهی هست که بتواند زمین‌ها را که متعلق به دو نفر هستند به شکلی بین آن‌ها تقسیم بکند که از زمین‌های یک نفر به طور مستقیم به زمین‌های دیگر آن شخص راهی نباشد. نیما از شما می‌خواهد این کار را برای او انجام دهید و براساس اطلاعات بدست آمده از منطقه، امکان‌پذیری این کار را مشخص کنید.

نیما اطلاعاتی در اختیار شما قرار می‌دهد که در ابتدا تعداد زمین‌ها و تعداد راه‌های موجود بین آن زمین‌ها را مشخص می‌کند. در ادامه با مشخص کردن این که تک تک راه‌ها کدام دو زمین را به هم وصل می‌کنند، از شما می‌خواهد تا مشخص کنید آیا اصلاً امکان تقسیم‌بندی زمین‌ها موجود هست یا نه.

## ورودی

در خط اول، تعداد زمین‌ها که با  $n$  نمایش داده می‌شود به عنوان ورودی داده شده و در خط دوم  $m$  که نشان‌دهنده تعداد راه‌هاست داده می‌شود

$$2 \leq n \leq 2000$$

$$0 \leq m \leq 2000$$

در  $m$  خط بعدی و در هر خط دو عدد آمده که نشان‌دهنده وجود یک مسیر بین آن دو زمین می‌باشد

## خروجی

در صورت برآورده شدن خواسته سوال عبارت Yes و در غیر این صورت No چاپ می‌شود.

## مثال

### ورودی نمونه ۱

3  
3  
1 0  
2 1  
0 2

### خروجی نمونه ۱

No

## کلمات و جدول

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

شما که در حال شرکت در یک مسابقه حل جدول بودید، تصمیم می‌گیرید تا با استفاده از نبوغ برنامه‌نویسی خود راه‌حلی سریع برای حل جدول کلمات انگلیسی پیدا کنید. شما تصمیم می‌گیرید تا برنامه‌ای بنویسید که در آن با ورودی گرفتن یک جدول، به دنبال کلمات داده شده در جدول بگردید. ابعاد جدول به شما داده می‌شود و در ادامه لیستی از کلمات به دنبال آن می‌آید. کلمات به این صورت پیدا می‌شوند که با شروع از یک خانه و رفتن به خانه‌های مجاور (به صورت خط و یا اریب) بتوانیم حرف به حرف کلمه را پیدا کنیم. در انتها به همان ترتیب ورودی کلمات، عبارت FOUND را در صورت پیدا شدن کلمه و یا NOT FOUND را در صورت نبود کلمه در جدول چاپ می‌کنید. دقت شود که لزومی ندارد که کلمات در یک سطر، ستون و یا قطر پیدا شوند. هدف این است که شما بررسی کنید که آیا می‌توان کلمه گفته شده را صرفاً با حرکت در یکی از جهت‌های ممکن (حداکثر ۸ جهت) بدون نیاز به هیچ پرشی پیدا کرد یا نه. (به عنوان مثال فرض کنید که با انگشت خود از یک خانه شروع کرده و با حرکت به جهت‌های مختلف و بدون برداشتن انگشت، کلمه مدنظر را پیدا می‌کنید).

## ورودی

در خط اول ورودی عدد  $N$  می‌آید. در ادامه و در  $N$  خط، رشته‌های  $N$  کاراکتری که محتویات درون جدول را نمایش می‌دهند داده می‌شوند. پس از آن عدد  $M$  در یک خط به صورت جداگانه ورودی داده می‌شود که این عدد تعداد کلماتی را که باید به دنبال آن بگردید را مشخص می‌کند. در  $M$  خط بعد هم کلمات به ترتیب و در خطوط جداگانه داده می‌شوند

$$3 \leq N, M \leq 30$$

## خروجی

به ترتیب ورودی کلمات، در صورت وجود کلمه در جدول عبارت FOUND و در صورت عدم وجود عبارت NOT FOUND نمایش داده می‌شود.

## مثال

### ورودی نمونه ۱

8

FBUFBFNY

BMZPNUVX

HJWAVDTN

HLBYTPT

VJDTYKHH

DAEGLGZP

KAKGSTNQ

KMDOGGRX

11

HGHRULBB

ERILG

HKYTDJV

EECOLWTD

HKYT

ASKZWI

ZOO

HTN

VQA

FPAYTGG

NTDVAWJ

### خروجی نمونه ۱

NOT FOUND

NOT FOUND

FOUND

NOT FOUND

FOUND

NOT FOUND

NOT FOUND

FOUND

NOT FOUND

FOUND

FOUND

## چاله چوله

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

هنگامی که باران زیادی در شهر شما می‌آید، به دلیل بد بودن وضعیت خیابان‌ها، آب زیادی در سطح معابر جمع می‌شود. شهرداری از شما خواسته است تا با استفاده از قدرت برنامه‌ریزی خود بتوانید فکری به حال این موضوع بکنید. با توجه به این که منابع شهرداری کم است، شما باید خیابان‌هایی که وضعیت بدتری داشته و همیشه آب بیشتری در آن جمع می‌شود را پیدا کنید و آن‌ها را در اولویت بیشتری برای تعمیر قرار دهید.

هدف شما این است تا با داشتن ارتفاع هر قسمت از آسفالت، مشخص کنید که به طور کلی چند واحد آب در این خیابان جمع می‌شود. داده شما به صورت یک ماتریس است که در هر خانه از آن، ارتفاع آن خانه نسبت به سطح صفر نوشته شده است. شما باید براساس این ماتریس مشخص کنید که در کل این خیابان چه مقدار آب باقی می‌ماند (فرض کنید هر واحد آب به اندازه یک واحد ارتفاع دارد). دقت شود که برای آن که یک آب بتواند در جایی جمع شود، باید چهار طرف آن به صورت کامل بسته باشد.

## ورودی

در خط اول، دو عدد  $n$  و  $m$  که نمایانگر ابعاد مستطیل هستند داده می‌شوند. سپس در  $n$  سطر بعد و در هر سطر  $m$  داده می‌شوند که هر کدام از این اعداد نمایانگر ارتفاع سطح زمین در آن خانه می‌باشند

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

## خروجی

صرفاً مقدار آب جمع شده در مستطیل داده‌شده چاپ می‌شود.

## مثال

## ورودی نمونه ۱

3 6  
1 4 3 1 3 2  
3 2 1 3 1 4  
2 3 3 2 3 1

## خروجی نمونه ۱

5

در این مثال اگر خانه بالا سمت چپ را (۰،۰) در نظر بگیریم، ۱ واحد در خانه (۱،۱)، دو واحد در (۱،۲) و دو واحد در (۱،۴) جمع می‌شوند.

## Logger

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

شما تصمیم گرفته‌اید تا یک برنامه برای مدیریت لاگ‌هایی که در یک سیستم ثبت می‌شوند را طراحی کنید. متوجه شده‌اید که قالب لاگ‌ها الگوی خاصی دارد که می‌توانید با استفاده از آن دستورات مختلف برای مدیریت لاگ‌ها اجرا کنید. فرمت لاگ‌ها به صورت زیر می‌باشد:

[YYYY-MM-DD HH:MM:SS] [LOG\_LEVEL] Message

در بخش اول تاریخ زمان آمدن لاگ نوشته شده است. در ادامه نوع لاگ مشخص می‌شود. انواع لاگ به صورت زیر می‌باشد:

1. INFO
2. WARNING
3. ERROR
4. DEBUG

در انتها نیز یک پیغام برای آن لاگ نوشته شده است که می‌تواند شامل هر کاراکتری باشد.

در ادامه از شما انتظار می‌رود تا دستورات مختلفی را برای مدیریت لاگ‌ها پیاده‌سازی کنید.

### بررسی نوع لاگ

با استفاده از دستور زیر شما باید تمام لاگ‌هایی که نوع لاگ مشخص شده در دستور را دارند جدا کرده و آن‌ها را به همان ترتیبی که در ورودی آمده است نمایش دهید:

LEVEL [LOG\_LEVEL]

## بازه زمانی لاگ‌ها

با استفاده از دستور زیر شما باید بتوانید تمام لاگ‌ها را، با همان ترتیبی که در ورودی آمده است، از بازه DATE1 و DATE2 نمایش دهید (فرمت تاریخ ها به صورت YY-MM-DD است):

```
DATE_RANGE [DATE1] [DATE2]
```

## نمایش لاگ‌های خطا

از آنجایی که لاگ‌هایی که نشان‌دهنده خطا هستند، شما می‌خواهید تاریخ تمام لاگ‌هایی که باعث خطا شده‌اند را نمایش دهید. دستور زیر چنین عملی را برای شما انجام می‌دهد:

```
ERROR_TIMESTAMPS [--reverse]
```

فلگ --reverse به صورت آپشنال است و وجود آن اجباری نیست. اگر این فلگ استفاده شود، لاگ‌هایی که باعث خطا شدند با ترتیب برعکس چاپ می‌شوند.

## کلمه خاص

گاهی اوقات شما به دنبال یک سری لاگ‌های خاص در مورد یک موضوع خاص می‌گردید. از این جهت، با استفاده از دستور زیر باید تمام لاگ‌هایی که شامل کلمه خاص داده شده در KEYWORD می‌شوند را نمایش دهید.

```
CONTAINS [KEYWORD]
```

## تعداد نوع لاگ‌ها

با استفاده از دستور زیر، شما باید تعداد لاگ‌هایی که از نوع [LEVEL] هستند را در خروجی چاپ کنید.

```
COUNT_LEVEL [LEVEL]
```

## تکرار کلمه

در این دستور شما می‌خواهید تا پرتکرارترین کلمات را پیدا کنید. هدف این است تا با بررسی قسمت message لاگ‌ها، پرتکرارترین کلمات را پیدا کنید. دستور به شکل زیر است:

```
FREQUENCY_ANALYSIS [--top] [number]
```

اگر دستور به صورت عادی و بدون فلگ --top و number استفاده شود، شما ۵ کلمه پرتکرار را نمایش می‌دهید. با آمدن --top و عدد در کنار آن، می‌توان مشخص کرد که چند کلمه پرتکرار نمایش داده شود.

## نمایش چند لاگ خاص

با استفاده از دستور زیر، شما باید بتوانید به تعداد مشخص شده لاگ از یک نوع خاص را که جدیدترین هستند را نمایش دهید. در واقع از شما خواسته شده است تا number از جدیدترین لاگ‌های یک نوع خاص را چاپ کنید. دقت شود که اگر تعداد کل لاگ‌ها از تعداد عدد داده شده کمتر است، کل لاگ‌های آن نوع چاپ می‌شوند. دستور به شکل زیر است:

```
TOP_K_LEVEL [number] [LOG_LEVEL] [--reverse]
```

فلگ --reverse که به صورت آپشنال است، این قابلیت را به ما می‌دهد تا به جای چاپ جدیدترین لاگ‌ها، قدیمی‌ترین لاگ‌ها را نمایش دهیم.

## ورودی

در خط اول ورودی عدد n می‌آید که نمایش‌دهنده تعداد لاگ‌هایی است که در ادامه خواهیم داشت. در n خط بعد و در هر خط، یک لاگ با فرمت گفته شده در بالا داده شده و در ادامه تعدادی دستور به عنوان ورودی داده می‌شود.

## خروجی

شما باید خروجی هر دستور را چاپ کنید و بین هرکدام از خروجی‌ها، یک خط خالی قرار دهید.

## مثال

### ورودی نمونه ۱

17

```
[2025-02-12 18:34:00] [INFO] System reboot required due to critical failure
[2025-02-11 21:00:00] [WARNING] High memory usage detected: 95% of available R
[2025-02-13 05:55:00] [DEBUG] Service restarted due to unexpected termination
[2025-02-12 04:51:00] [INFO] Memory allocation failure during intensive comput
[2025-02-11 14:24:00] [INFO] Unauthorized access attempt detected from IP 203.
[2025-02-12 01:02:00] [WARNING] Disk write error encountered while saving crit
[2025-02-13 04:38:00] [INFO] User session expired after prolonged inactivity
[2025-02-11 01:44:00] [DEBUG] Process started successfully with PID 3579
[2025-02-10 20:28:00] [DEBUG] Memory allocation failure during intensive compu
[2025-02-12 08:34:00] [INFO] Unauthorized access attempt detected from IP 203.
[2025-02-11 07:56:00] [INFO] Disk write error encountered while saving critica
[2025-02-11 12:49:00] [ERROR] User logged in successfully from IP 192.168.1.10
[2025-02-12 02:41:00] [ERROR] Disk write error encountered while saving critic
[2025-02-13 07:41:00] [WARNING] Process started successfully with PID 3579
[2025-02-12 10:49:00] [WARNING] Service restarted due to unexpected terminatic
[2025-02-11 15:27:00] [WARNING] System reboot required due to critical failure
[2025-02-11 08:46:00] [INFO] User logged in successfully from IP 192.168.1.10
COUNT_LEVEL INFO
LEVEL DEBUG
CONTAINS Process
CONTAINS System
ERROR_TIMESTAMPS --reverse
LEVEL ERROR
TOP_K_LEVEL 2 WARNING
ERROR_TIMESTAMPS
TOP_K_LEVEL 2 WARNING --reverse
FREQUENCY_ANALYSIS --top 3
COUNT_LEVEL ERROR
```

### خروجی نمونه ۱

7

[2025-02-13 05:55:00] [DEBUG] Service restarted due to unexpected termination  
[2025-02-11 01:44:00] [DEBUG] Process started successfully with PID 3579  
[2025-02-10 20:28:00] [DEBUG] Memory allocation failure during intensive compu

[2025-02-11 01:44:00] [DEBUG] Process started successfully with PID 3579  
[2025-02-13 07:41:00] [WARNING] Process started successfully with PID 3579

[2025-02-12 18:34:00] [INFO] System reboot required due to critical failure  
[2025-02-11 15:27:00] [WARNING] System reboot required due to critical failure

2025-02-12 02:41:00  
2025-02-11 12:49:00

[2025-02-11 12:49:00] [ERROR] User logged in successfully from IP 192.168.1.10  
[2025-02-12 02:41:00] [ERROR] Disk write error encountered while saving critic

[2025-02-13 07:41:00] [WARNING] Process started successfully with PID 3579  
[2025-02-12 10:49:00] [WARNING] Service restarted due to unexpected terminatic

2025-02-11 12:49:00  
2025-02-12 02:41:00

[2025-02-11 15:27:00] [WARNING] System reboot required due to critical failure  
[2025-02-11 21:00:00] [WARNING] High memory usage detected: 95% of available R

to: 7  
critical: 5  
due: 4

2