

## رمزنگاری

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

جواد روشی برای رمزنگاری اختراع کرده است. به این صورت که هر بار حرف وسط رشته را یادداشت می‌کند و آن را از رشته حذف می‌کند تا طول رشته صفر شود. (اگر طول رشته زوج بود، حرفی که اندیس آن **کمتر** است در نظر گرفته می‌شود) همچنین، هر بار که طول رشته **فرد** می‌شود، آن را کاملاً برعکس می‌کند و عملیات را روی آن ادامه می‌دهد.

رشته‌ای که جواد می‌خواهد آن را رمزنگاری کند، به شما داده می‌شود. شکل رمزنگاری شده‌ی آن را بنویسید.

## ورودی

شامل یک خط است که در آن یک رشته آمده است.

## خروجی

رشته‌ی رمزنگاری شده توسط جواد چاپ می‌شود.

## مثال

## ورودی نمونه ۱

football

## خروجی نمونه ۱

tbaoollf

دنباله‌ی مراحل بدین صورت است:

رشته	یادداشت
football	
fooball	t
llaoof	tb
lloof	tba
foll	tbao
flf	tbao

رشته	یادداشت
lf	tbaool
f	tbaooll
	tbaoollf

ورودی نمونه ۲

salamJAVAd

خروجی نمونه ۲

mJAa1VAasd

## عملیات

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

با دریافت رشته‌ای بدون فاصله از حروف انگلیسی، اعداد و علائم، عملیات زیر را روی آن انجام دهید:

۱. تمام کاراکترهای رشته به جز **حروف و اعداد** را حذف کنید.
۲. **طول** بلندترین زیر رشته‌ی *palindrome* (از هر دو طرف یکسان خوانده می‌شود) را برای رشته‌ی حاصل از عملیات قبل به‌دست آورید.

## ورودی

در یک خط، رشته با شرایط گفته شده داده می‌شود، با این تضمین که شامل حداقل یک حرف یا عدد می‌باشد.

## خروجی

در خط اول رشته‌ی حاصل از عملیات اول و در خط بعد، عدد حاصل از عملیات دوم را چاپ کنید.

## مثال

## ورودی نمونه ۱

```
"Ali"Salam!
```

## خروجی نمونه ۱

```
AliSalam
3
```

رشته‌ی "ala"، طولانی‌ترین زیر رشته‌ی palindrome است.

## ورودی نمونه ۲

```
?ab?cd?
```

## خروجی نمونه ۲

```
abcd
1
```

## جابه‌جایی

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

رشته‌ی اصلی  $s$  و دو رشته‌ی  $u$  و  $v$  شامل حروف کوچک انگلیسی در ورودی داده می‌شوند.

**اولین** وقوع  $v$  را با **آخرین** وقوع  $u$  در رشته‌ی اصلی جابه‌جا کنید و رشته‌ی حاصل را در خروجی نمایش دهید.

اگر رشته‌ی  $s$  **فقط** شامل  $v$  باشد، تنها،  $u$  را جایگزین اولین وقوع  $v$  می‌کنیم و اگر **فقط** شامل  $u$  باشد، کافی است  $v$  را جایگزین آخرین وقوع  $u$  کنیم؛ و اگر  $s$  شامل **هیچ** یک از دو رشته‌ی دیگر نباشد، خود  $s$  چاپ می‌شود.

## ورودی

در خط اول رشته‌ی اصلی  $s$ ، در خط دوم رشته‌ی  $u$  و در خط سوم رشته‌ی  $v$  می‌آیند.

$$1 \leq |s|, |u|, |v| \leq 10000$$

## خروجی

رشته‌ی جدید را چاپ کنید.

## مثال

### ورودی نمونه ۱

abcdeabcd  
ab  
cd

### خروجی نمونه ۱

ababecdcd

### ورودی نمونه ۲

ab  
ab  
cd

### خروجی نمونه ۲

cd

آخرین و تنها وقوع رشته‌ی ab، با رشته‌ی cd جایگزین شده‌است.

## ورودی نمونه ۳

ab  
cd  
e

## خروجی نمونه ۳

ab

رشته‌ی  $s$  شامل هیچ یک از دو رشته‌ی دیگر نیست، پس خود آن چاپ می‌شود.



## جدول

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک جدول  $m$  در  $n$  از حروف کوچک انگلیسی به شما داده می‌شود. آن را فقط به صورت **افقی** و **عمودی** بررسی کنید و تعداد کلمات معنادار آن را به دست آورید.

کلمه‌ی معنادار، کلمه‌ای است که:

۱. طول بیش‌تر از یک داشته باشد.
۲. دو حرف صد/دار متوالی نداشته باشد.
۳. دو حرف یکسان متوالی نداشته باشد.

## ورودی

ابتدا دو عدد  $m$  و  $n$  در ورودی می‌آیند که به‌ترتیب نشان‌دهنده‌ی تعداد سطرها و ستون‌های جدول هستند. سپس در  $m$  خط بعدی، در هر خط،  $n$  حرف داده می‌شود.

$$1 \leq n, m \leq 1000$$

## خروجی

تعداد کلمات معنادار جدول را در یک خط خروجی چاپ کنید.

## مثال

### ورودی نمونه ۱

```
3 4
a e b b
m i b b
m u u o
```

### خروجی نمونه ۱

8

کلمات معنادار این جدول عبارت‌اند از :

*eb, mi, mib, ib, mu, am, bu, bo*

## ایمیل

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

یک ایمیل در قالب کلی زیر ساخته می‌شود:

1 | [local-part]@[domain-name].[domain]

- هیچ‌کدام از قسمت‌های آن نباید خالی باشند.
  - قسمت *domain* فقط شامل حروف کوچک و بزرگ انگلیسی می‌باشد.
  - قسمت‌های *local-part* و *domain-name* نمی‌توانند شامل @ یا علائم نگارشی باشند.
  - در اینجا علائم نگارشی ، ، ، : و ! هستند.
- حال با دریافت یک رشته به شکل email:text تعداد تکرارهای email را در text به دست آورید.

## ورودی

رشته‌ی به شکل بالا، شامل email و text داده می‌شود.

## خروجی

در صورت نامعتبر بودن email، مقدار 1- و در غیر این صورت، تعداد تکرارهای آن در text را چاپ کنید.

## مثال

### ورودی نمونه ۱

```
ab@cd.ir:ab@cd.ir is the same as ab@cd.ir, and different from: z@cde.com and sab@cd.ir but same
as:ab@cd.ir
```

### خروجی نمونه ۱

```
3
```

### ورودی نمونه ۲

```
ali@alavi.ac.:salam
```

### خروجی نمونه ۲

```
-1
```

قسمت‌های مختلف email در مثال داده‌شده به شکل زیر است:

- قسمت local-part: همان ali
- قسمت domain-name: همان alavi.ac
- قسمت domain: خالی است.

## اسامی خاص

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

برنامه‌ای بنویسید که کلیه اسامی خاص موجود در یک متن انگلیسی را چاپ کند. تعریف از اسامی خاص، کلمه‌هایی است که در اول جمله قرار **ندارند** ولی با حرف بزرگ شروع می‌شوند. به نکات زیر دقت کنید:

- هر اسم خاص، حداقل ۲ حرف دارد.
- کلمات باید به ترتیب وقوع چاپ شوند.
- جملات با **نقطه** یا **علامت سوال** به اتمام می‌رسند.
- بین هر دو جمله یک کاراکتر فاصله قرار دارد.

## ورودی

شامل یک خط است که در آن جملات ورودی آمده‌اند.

## خروجی

کلمات خاص ورودی که هر یک، در یک خط جدا چاپ شده‌اند.

## مثال

### ورودی نمونه ۱

Sir Daniel Michael Blake Day-Lewis (born 29 April 1957) is an English actor who holds both British and Irish citizenship. Born and raised in London, he excelled on stage at the National Youth Theatre, before being accepted at the Bristol Old Vic Theatre School, which he attended for three years.

### خروجی نمونه ۱

Daniel  
Michael  
Blake  
Day-Lewis  
April  
English  
British  
Irish  
London  
National  
Youth  
Theatre  
Bristol  
Old  
Vic

Theatre  
School

متن ورودی شامل دو جمله است.



## رتبه‌بند

- محدودیت زمان ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه‌ای بنویسید که مجموعه‌ای از اسناد را از نظر مرتبط بودن با یک واژه‌ی خاص رتبه‌بندی کند.

- هر سند مجموعه‌ای از کلمات است که به واسطه‌ی تعدادی فاصله یا علامت‌های ، یا ! یا ؟ یا . از هم جدا شده‌اند.
- ارتباط هر سند با واژه‌ی خاص از رابطه‌ی  $X * Y$  بدست می‌آید.  $X$  تعداد کل واژه‌های موجود در سند است و  $Y$  تعداد کلماتی که اگر پیشوندی به طول  $prefix$  از آن‌ها جدا کرده و آن را  $p$  بنامیم و پسوندی به طول  $postfix$  از آن‌ها جدا نموده و آن را  $q$  بنامیم، واژه‌ی خاص دارای پیشوند  $p$  و پسوند  $q$  باشد. بزرگ یا کوچک بودن حروف، تاثیری در شباهت کلمات ندارد.

متد زیر را با توجه به توضیحات داده شده پیاده‌سازی کنید:

```
1 | static int[] rank(String word, int prefix, int postfix, String[] documents)
```

- رشته‌ی  $word$ : کلمه‌ای که ارتباط اسناد با آن بررسی می‌شود.
- عدد  $prefix$ : **تعداد** حروف ابتدای کلمه که برای بررسی پیشوندها استفاده می‌شود.
- عدد  $postfix$ : **تعداد** حروف انتهای کلمه که برای بررسی پسوندها استفاده می‌شود.
- آرایه‌ی  $documents$ : اسناد مورد بررسی که هریک از جنس رشته می‌باشند.

- خروجی متد، یک آرایه از جنس *int* شامل اندیس اسناد مرتب شده از نظر ارتباط با *word* است. مرتب سازی به صورت **صعودی** انجام می شود و در صورت برابری رتبه ی دو سند، سندی که در ورودی زودتر آمده است، در خروجی هم زودتر می آید.

## مثال:

در تکه کد زیر، ۸ سند داده شده است. با فراخوانی متد *rank* اسناد به ترتیب ارتباطشان با کلمه ی *salam*، رتبه بندی می شوند.

```
1 String[] documents = new String[8];
2 documents[0] = "hola hi hello salam? hello."; // y=1, x=5, rank = 5
3 documents[1] = "hola hola hello,hello hi! hi."; // y=0, x=6, rank = 0
4 documents[2] = "salam salam hi! hola hello hello."; // y=2, x=6, rank = 12
5 documents[3] = "salam hi hola salam salam hi."; // y=3, x=6, rank = 18
6 documents[4] = "salam hi. hi! hi salam hi hola."; // y=2, x=7, rank = 14
7 documents[5] = "salam hi. hi! hi hi hi hola."; // y=1, x=7, rank = 7
8 documents[6] = "salam hi. hi! salad salaam saalaaam hola salaaaa sallam."; // y=4, x=9, rank = 36
9 documents[7] = "salam hi. hi! sos salaam saalaaam hola."; // y=3, x=7, rank = 21
10 int[] indices = rank("salam", 1, 1, documents);
11 System.out.println(Arrays.toString(indices));
12 // expected: [1, 0, 5, 2, 4, 3, 7, 6]
```

## ورودی

در خط اول *n* و *word* و *p* و *q* داده می شوند که به ترتیب، تعداد اسناد، کلمه ی موردنظر برای بررسی ارتباط، مقدار *prefix* و مقدار *postfix* هستند. در *n* خط بعدی، در هر خط یک سند داده می شود.

$$1 \leq n \leq 100$$

$$1 \leq |word| \leq 100$$

$$0 \leq p, q \leq |word|$$

## خروجی

آرایه‌ی اندیس‌های اسناد مرتب شده که مطابق مثال چاپ می‌شود.

## مثال

### ورودی نمونه ۱

```
8 salam 1 1
hola hi hello salam? hello.
hola hola hello,hello      hi! hi.
salam salam hi! hola      hello hello.
salam hi hola salam salam hi.
salam hi. hi! hi salam hi hola.
salam hi. hi! hi hi hi hola.
salam hi. hi! salad salaam saalaaam hola salaaaa sallam.
salam hi. hi! sos salaam saalaaam hola.
```

خروجی نمونه ۱

[1, 0, 5, 2, 4, 3, 7, 6]

ورودی نمونه ۲

2 abcde 3 3  
abccde abcde abcdefcde  
abcd adcbe a!? aabcde

خروجی نمونه ۲

[1, 0]

## زیرالفبا

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

برای یک رشته، **زیرالفبا** مجموعه‌ی کاراکترهای **متفاوت** موجود در آن است. برای مثال اگر رشته `XhTt2ZLL` باشد، زیرالفبای آن برابر با  $\{X, H, T, 2, Z, L\}$  خواهد بود. برای تعیین زیرالفبا، بین حروف کوچک و بزرگ، تفاوتی قائل **نمی‌شویم**.

ابتدا رشته‌ی  $T$  و سپس دنباله‌ی  $s_1, s_2, \dots, s_n$  از رشته‌ها در ورودی داده می‌شود. زیرالفبای رشته‌ی  $T$  را مجموعه‌ی  $t$  می‌نامیم. اگر زیرالفبای رشته‌ای **دقیقا** با مجموعه‌ی  $t$  برابر بود، آن رشته را **صحیح** می‌نامیم.

## ورودی

سطر اول ورودی شامل عدد طبیعی  $n$  و رشته‌ی  $T$  است. سپس در  $n$  سطر بعدی به ترتیب  $s_1$  و  $s_2$  و ... و  $s_n$  آمده‌است. تضمین می‌شود رشته‌های ورودی تنها از حروف کوچک و بزرگ و ارقام انگلیسی تشکیل شده‌اند.

$$1 \leq n \leq 100$$

$$1 \leq |s_i|, |t| \leq 100$$

## خروجی

در خروجی باید  $n$  سطر چاپ شود. اگر  $s_i$  رشته‌ای صحیح بود، در سطر متناظر آن Yes و در غیر این صورت No چاپ کنید.

## مثال

### ورودی نمونه

```
4 quera102
quEra0012
qu0erraa12
sN0Ap12
qurra00L
```

### خروجی نمونه

```
Yes
Yes
No
No
```

زیرالفبای رشته‌های ورودی بدین صورت است:

$$quera102 = \{Q, U, E, R, A, 0, 1, 2\}$$

$$quEra0012 = \{Q, U, E, R, A, 0, 1, 2\}$$

$$qu0erraa12 = \{Q, U, E, R, A, 0, 1, 2\}$$

$$sN0Ap12 = \{S, N, A, P, 0, 1, 2\}$$

$$qurra00L = \{Q, U, R, A, L, 0\}$$

## معادله (امتیازی)

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک معادله به فرم کلی  $A ? B = C$  به شما داده می‌شود.  $A$  و  $B$  و  $C$  اعداد **صحیح مثبت** و بدون صفر پشت عدد هستند و علامت سوال، به جای یکی از عملگرهای اصلی  $+$ ,  $-$ ,  $*$ ,  $/$  می‌آید.

**یکی** از سه عدد  $A$  و  $B$  و  $C$  شامل علامت  $\#$  است؛ این علامت جایگزین  $x$  رقم متوالی از آن عدد است با این فرض که طول  $x$  می‌تواند حداقل صفر و حداکثر به اندازه‌ی طول عدد باشد.

بررسی کنید که به ازای هر عملگر، چه ارقامی به جای  $\#$  قرار گیرند تا یک معادله‌ی درست ساخته شود، سپس تمام معادلات صحیح را چاپ کنید. اگر به ازای یک عملگر، معادله قابل ساخت نبود، معادله‌ی مربوط به آن را چاپ نکنید.

## ورودی

در یک خط، معادله به شکل  $A ? B = C$  داده می‌شود که فقط یکی از اعداد آن شامل  $\#$  است.

$$1 \leq A, B, C \leq 10^9$$

## خروجی



در صورت امکان قرار دادن عدد به جای # و عملگر به جای ؟، معادلات صحیح حاصل از جایگذاری را به ترتیب برای عملگرهای + و - و \* و / چاپ کنید. اگر هیچ معادله‌ای پیدا نکردید، 1- چاپ کنید.

## مثال

### ورودی نمونه ۱

10 ? 2 = #

### خروجی نمونه ۱

10 + 2 = 12

10 - 2 = 8

10 \* 2 = 20

10 / 2 = 5

### ورودی نمونه ۲

10#2 ? 20 = 10022

### خروجی نمونه ۲

$$10002 + 20 = 10022$$

$$10042 - 20 = 10022$$

فقط به ازای دو عملگر + و - ، معادله‌ی صحیح داریم.

ورودی نمونه ۳

$$11 \ ? \ 15 = \#2$$

خروجی نمونه ۳

$$-1$$

ورودی نمونه ۴

$$23 \ ? \ \# = 1$$

خروجی نمونه ۴

$$23 - 22 = 1$$

$$23 / 23 = 1$$

به ازای عملگر + ، عدد مجهول منفی و به ازای عملگر \* ، عدد مجهول اعشاری می شود که قابل قبول نیست.

## محاسبه گر (امتیازی)(C++)

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

برنامه‌ای بنویسید که یک عبارت ریاضی گرفته و حاصل آن را حساب کند.  
عبارت ریاضی شامل ضرب (\*)، تقسیم (/)، توان (^)، جمع (+)، تفریق (-) و پرانتز است و بین اعداد و عملگرها می‌تواند به تعداد دلخواه فاصله باشد!

### ورودی

در خط اول یک عبارت ریاضی است.

### خروجی

جواب عبارت ریاضی را با **دو رقم اعشار** چاپ کنید

### مثال

ورودی نمونه ۱

$$1+2*(5-3)/4+6$$

خروجی نمونه ۱

$$8.00$$

ورودی نمونه ۲

$$(3 \wedge 2) * 5 + 2$$

خروجی نمونه ۲

$$47.00$$