

پردازش تصویر (Kiroshi Optics)

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



در آینده Cyberpunk 2077، اکثر افراد به جای چشم‌های طبیعی‌شان، از چشم‌های مصنوعی استفاده می‌کنند. این تکنولوژی می‌تواند حتی به افراد نابینا نیز توانایی بینایی را برگرداند! از سودها و ثمرات استفاده از این چشمان مصنوعی امکان دید شب، زوم کردن برای دید بهتر اشیای دور، دیدن اشعه‌های حرارتی محیط، رفع کور رنگی و ... می‌باشد. شما هم که این چشمان مصنوعی را دارید، کنجکاو می‌شوید که بدانید چگونه این تکنولوژی، توانایی این را دارد که این گونه عکس‌ها و شکل‌های محیط‌های پیرامون شما را عوض کند و به اعصاب شبکیه شما بفرستد. چون شما خیلی علاقه به زیست‌شناسی ندارید، ترجیح می‌دهید از دید کامپیوتری درباره این موضوع تحقیق کنید...

می‌خواهیم عملیاتی را پیاده‌سازی کنیم که در پردازش تصویر و یادگیری ماشین بسیار پرکاربرد است و به آن **کانولوشن دو بعدی** (Two-Dimensional Convolution) گفته می‌شود.

فرض کنید یک ماتریس عددی داریم که می‌تواند نمایندهٔ یک تصویر یا دادهٔ دوبعدی باشد. همچنین یک ماتریس کوچک‌تر به نام **فیلتر** (یا کرنل) در اختیار داریم.

ایدهٔ اصلی کانولوشن این است که این فیلتر، روی بخش‌های مختلف ماتریس ورودی حرکت می‌کند. در هر موقعیت، فیلتر روی ناحیه‌ای از ماتریس قرار می‌گیرد و با آن ناحیهٔ ترکیب می‌شود:

هر **درایه** از فیلتر در **درایهٔ متناظر** خودش از آن بخش از ماتریس **ضرب** می‌شود، و سپس **جمع همهٔ** این **حاصل ضرب‌ها** یک **درایهٔ جدید** می‌سازد. این **درایه**، نشان‌دهندهٔ نتیجهٔ فیلتر در همان موقعیت است.

با تکرار این فرآیند در سراسر ماتریس، مجموعه‌ای از **درایه‌های جدید** به دست می‌آید که در کنار هم، ماتریس **جدیدی** را تشکیل می‌دهند که همان **خروجی کانولوشن** است.

برای کنترل نحوهٔ حرکت فیلتر، دو پارامتر وجود دارد:

- **پدینگ (Padding)**: تعداد سطر یا ستونی از صفرها که دور ماتریس ورودی اضافه می‌شود تا فیلتر بتواند روی لبه‌ها نیز قرار بگیرد.
- **گام (Stride)**: نشان می‌دهد فیلتر پس از هر بار محاسبه، چند **درایه سطري** یا ستونی روی ماتریس جلو ببرود.

برای درک بهتر این عملیات، پیشنهاد می‌شود [این ویدیو](#) را تماشا کنید.

یکی از مثال‌های کاربرد این عملیات، **Blur** کردن تصویر است که نمونه‌ای از آن در شکل زیر مشخص است.

original



convolved



وروُدی

ابتدا دو عدد m و n وارد می‌شوند که به ترتیب نشان‌دهنده تعداد سطرها و ستون‌های ماتریس ورودی هستند. سپس یک ماتریس $m \times n$ وارد می‌شود. پس از آن دو عدد k و p وارد می‌شوند که به ترتیب نشان‌دهنده تعداد سطرها و ستون‌های فیلتر هستند و سپس ماتریس فیلتر نیز به همان ترتیب وارد می‌شود. در ادامه، دو عدد دیگر وارد می‌شود: مقدار پدینگ و مقدار گام. تمامی مقادیر ورودی را در محدوده int بگیرید.

خروجی

شما باید ماتریس حاصل را در خروجی چاپ کنید.

ورودی نمونه ۱

```
6 4
-2 -3 3 7
6 1 7 1
3 -4 1 -3
5 1 10 5
3 -7 -8 8
10 -8 -6 -2
2 2
-5 5
-8 10
0
1
```

خروجی نمونه ۱

```
-43 92 -26
-89 72 -68
-65 117 -50
-114 21 119
-210 -1 108
```

ورودی نمونه ۲

9 8
10 -6 0 7 -4 1 -8 4
-10 -8 -4 -7 8 -7 0 10
-3 -8 -4 -3 -9 4 8 3
6 5 -3 7 -6 9 2 -1
1 -9 -4 5 4 -1 1 9
-2 3 5 7 9 8 -3 -5
-5 4 8 -3 1 0 9 -8
-2 10 10 -1 9 -10 -1 0
2 -4 10 -2 -5 3 6 -9
1 7
-4 -7 -8 9 -6 9 7
0
2

خروجی نمونه ۲

42
219
110
-42
21

ورودی نمونه ۳

12 19
-3 -7 -5 -10 -5 -7 10 7 3 -1 10 -8 -5 4 10 7 10 -10 -7
-8 0 -3 8 -3 0 -2 -8 5 3 -3 -3 2 2 -7 8 -10 -7 -9
8 9 4 -5 -4 -10 -10 -4 7 -1 2 0 -10 -7 2 5 4 3 8
8 -4 4 4 0 8 -7 -4 -10 -10 -3 8 3 3 -6 -10 6 7 0
-1 9 7 -2 -1 4 -6 -1 -1 -1 5 -1 -2 8 -5 3 -1 4 -3
8 -2 -2 -1 3 -1 5 -5 6 3 -2 -3 1 9 9 -10 -3 -9 -10
-9 10 -6 1 9 -4 4 9 -1 -6 9 -5 10 -10 8 9 1 6 4
-7 -5 6 1 2 -9 6 -6 7 -9 0 4 7 10 -4 6 5 -6 -10
-8 8 5 -5 -1 10 8 8 6 -8 -2 10 -3 -5 -5 9 7 -3 -2
8 5 1 -9 6 7 -2 10 -10 -7 -6 4 -4 3 -2 4 -1 6 -4
-5 3 -1 4 2 -6 -3 -6 4 2 1 -7 7 -7 10 -5 8 -3 -8
9 -7 9 4 0 -5 -1 4 8 -2 -6 -9 -9 -2 10 -9 -8 4 1
4 5

```
10 1 -2 8 3  
-5 2 -6 9 8  
-10 8 3 4 2  
2 3 1 7 6  
3  
3
```

خروجی نمونه ۳

```
-63 -132 70 97 9 121 -15  
-35 -4 -235 55 9 101 -27  
140 22 26 -96 -32 168 -112  
-75 123 318 -238 207 298 -31  
80 15 -190 -13 -161 -42 24
```

رمز دیوار سیاه (Blackwall Cipher)

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



نتواج (NetWatch) یک سازمان بین‌المللی است که در سال 2013 تا به حال حاضر (2077) وظیفه خطیر حفاظت سایبری و محافظت از بشریت در برابر هوش‌های مصنوعی فرا خطرناک را به عهده دارد. یکی از بزرگ‌ترین وظایف آن‌ها ممانعت از عبور هوش‌های مصنوعی خطرناک یا انسان‌ها، بین دو محدوده‌ی امن و حفاظت‌نشده‌ی فضای سایبری است. این دو بخش توسط یک دیوار قدرتمند به نام **دیوار سیاه** (The Blackwall) از هم جدا شده‌اند. تلاش برای عبور از این دیوار یک جرم بین‌المللی محسوب می‌شود.



نتواج تعدادی هکر که قصد عبور از دیوار را دارند را زیر نظر دارد و طی زمان می‌فهمد که این هکرها برخلاف ظاهر ترسناکشان، آن قدر باهوش نیستند که پیام‌های خود را با یک روش امن رمزگاری کنند! معلوم می‌شود که آنها از روش ساده **رمز سزار** برای رمزگاری استفاده می‌کنند. از آنجایی که شما یک برنامه‌نویس خبره، ولی آنچنان سرشناس نیستید، نتواج از شما درخواست می‌کند که یک برنامه سبک برای شکستن

رمزنگاری پیامهای این هکرها بنویسید. از آنجایی که شما قصد ساختن یک رزومه قابل قبول برای خودتان را دارید، با شوق و اشتیاق فراوان می‌پذیرید.

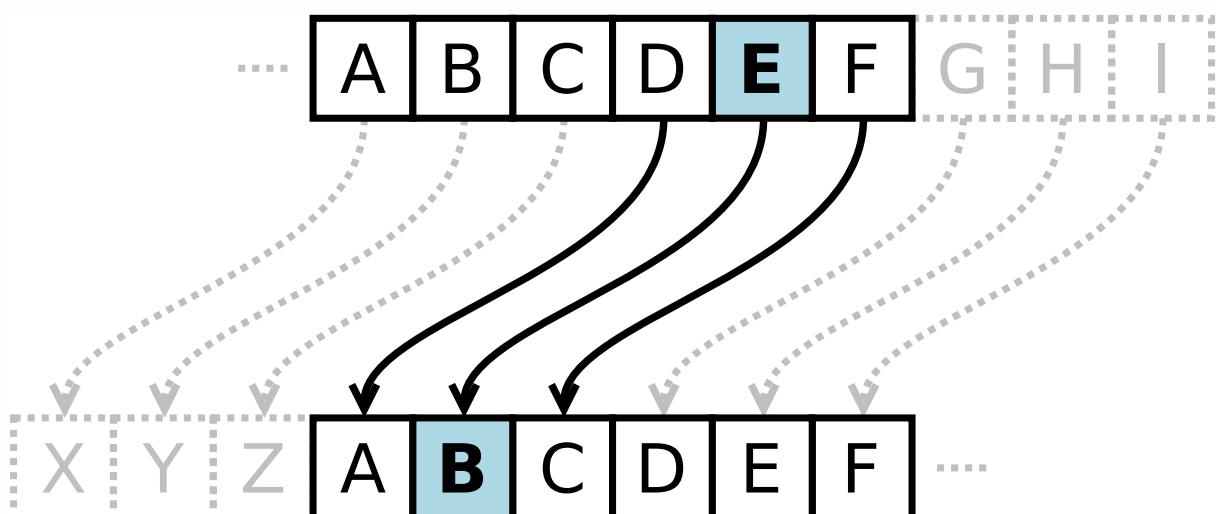
رمز سزار، در این تکنیک رمزنگاری، همه حروف متنی که می‌خواهیم رمزنگاری کنیم را به اندازهٔ یک عدد ثابت و به صورت دایره‌ای (دوری) جابه‌جا می‌کنیم. از آنجایی که زبان انگلیسی ۲۶ حرف دارد، هر متن را می‌توان حداقل به ۲۵ شیوه رمز کرد، زیرا جابه‌جایی‌های بیشتر از ۲۶ سبب می‌شوند، که حروف، بیش از یک مرتبه کل الفبا را دور بزنند. به عبارتی، باقی‌ماندهٔ جابه‌جایی بر ۲۶ معیار خواهد شد. به عنوان مثال، جابه‌جایی با مقدار ۳، حرف A را به D، و حرف Z را به C تبدیل می‌کند و سایر حروف نیز به همین شیوه رمزنگاری می‌شوند.

بدین‌گونه، اگر تابع دو متغیره‌ی caesarShift(letter, shift) را معرفی کنیم، خواهیم داشت:

$$\Sigma = \{a, b, c, \dots, z\} \cup \{A, B, C, \dots, Z\}$$

$$\text{caesarShift}: \Sigma \times \mathbb{N} \rightarrow \Sigma$$

$$\text{caesarShift}(x, s_1) = \text{caesarShift}(x, s_2) \text{ iff } s_1 = s_2 \bmod 26$$



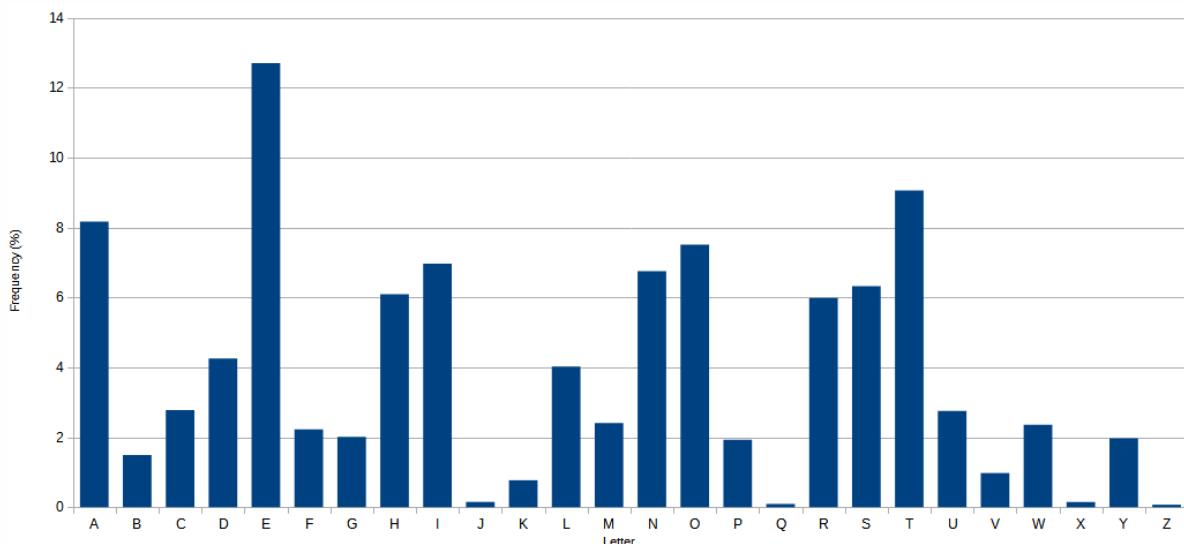
*متن نمونه

Ciphertext: Wklv lv dq hqfubswhg phvvdjh. Dg Ylfwruld! (Shift = 3)

Original Text: This is an encrypted message. Ad Victoriam!

لازم به ذکر است، که کاراکترهای غیرانگلیسی (مانند نقطه، ویرگول و فواصل) تحت تاثیر این رمز بی‌تغییر هستند. همچنین، حروف بزرگ (Capital) تحت این رمز بزرگ می‌مانند. مشابه این موضوع برای حروف کوچک نیز (Lowercase) برقرار است.

از آنجایی که 25 شیوه برای رمزگاری سزاری هر متن بیشتر نداریم، می‌توان با Bruteforce (بررسی همه ۲۵ حالت) متوجه شد که متن واقعی چه بوده است. به قدری شیفت را تغییر می‌دهیم تا یک متن مثل سایتهايي بياوريم. بعدياً معنا نظر از فهم قابل است. ابزار کاربردی‌ای برای شکستن این شیوه <https://www.boxentriq.com/code-breaking/caesar-cipher> ساده‌ی رمزگاری هستند. از آنجایی که در درس مبانی برنامه‌سازی، ما امکان شناسایی متن توسط هوش مصنوعی را نداریم، از روش دیگری برای شکستن رمز سزار استفاده می‌کنیم. در بررسی‌های آماری زبان انگلیسی، نموداری موسوم به **نمودار فراوانی** یا **فرکانس** برای هر حرف این زبان به وجود آمده است. این نمودار، فراوانی کاربرد، یا ظاهر شدن هر حرف را روی تعداد بسیار زیادی متن تجمعی کرده است.



همان‌طور که می‌بینیم، فراوان‌ترین حرف در زبان انگلیسی، E است و حدوداً 12.5 درصد هر متن متوسط را تشکیل می‌دهد. با فرض اینکه این درصد کاربرد برای یک متن مورد رمزگاری برقرار باشد، با تحلیل فرکانسی متن رمز شده و پیدا کردن حرفی که فراوانی اش حدوداً در این بازه باشد، می‌توانیم بگوییم که حرف E طی رمزگاری به چه حرف دیگری تبدیل شده و مقدار شیفت کاراکترها را این‌گونه پیدا کنیم!

* تکته: این روش لزوماً دقیق نیست، اولاً، درباره‌ی متن‌های کوچک، این تحلیل فرکانسی صحیح نیست و می‌تواند دارای خطأ باشد. چرا که فضای نمونه‌ی آماری ما بسیار کوچک است. همچنین، تعریف اینکه بازه‌ی فرکانس چه مقادیری باشد، نیز دقیق نیست. برای اجتناب از پیچیدگی بی‌مورد مسئله، ما این موارد را در تست‌ها و نحوه‌ی خروجی در نظر گرفته‌ایم.

ورودی

یک متن انگلیسی رمز شده (نه لزوما با رمز سزار!) در یک خط به شما داده می شود. تضمین می شود طول متن حداقل 500 کاراکتر باشد و متن خالی نیز نباشد.

خروجی

متن داده شده را تحلیل فرانسی کنید و اگر بیشینه‌ی فرانس در بازه‌ی $[10.0, 14.0]$ درصد بود، آن حرف را شیفت یافته‌ی E فرض کنید. سپس متن فوق را به عنوان یک رمز سزار بشناسید. تضمین می شود تنها یک حرف بیشینه فرانس را داشته باشد.

در خط نخست خروجی، عدد شیفت را چاپ کنید. توجه کنید که عدد شیفت در بازه‌ی 0 الی 25 است. در خط دوم متن اصلی را چاپ کنید. برای این کار، می‌توانید به وسیله‌ی شیفت معکوس دادن هر حرف، رمز را رمزگشایی کنید. اگر فرانس بیشینه در بازه‌ی اخیر نبود، این متن را در خروجی چاپ کنید:

Ciphertext is either too short or not a Caesar cipher!

ورودی نمونه ۱

Tevsec Mkockb gkc k Bywkx qoxobkv, cdkdocwkx, kxn gbsdob gry bocrkzon Bywkx

خروجی نمونه ۱

10

Julius Caesar was a Roman general, statesman, and writer who reshaped Roman

ورودی نمونه ۲

Oiahtr Bfshzq vzx o Gnlzm lscdqzk, xhpsdrlf, pmc vqnhtq vgn wshgzodi Fdlzm

خروجی نمونه ۲

Ciphertext is either too short or not a Caesar cipher!

در صورت علاقهمندی بیشتر، میتوانید بررسی کنید که متن بالا با رمز **ویژنر** رمز شده است!
(Vigenere Cipher)

ورودی نمونه ۳

Esp N aczrclxxtyr wlyrfllrp hld opgpwzap ty esp plcwj 1970d mj Opyytd Ctens

خروجی نمونه ۳

11

The C programming language was developed in the early 1970s by Dennis Ritchie

(Relic Engram Compression) فشرده‌سازی فوق پیشرفته

- محدودیت زمان: ۱۰ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



در دنیای خیالی و فوق پیشرفته Cyberpunk 2077، پیشرفت تکنولوژی به جایی رسیده است که می‌توان از مغز و شخصیت یک انسان یک کپی کامل و برابر اصل گرفت و آن را بر روی یک بیو-ترشه (Biochip) قرار داد! این تکنولوژی که موسوم به «عتیقه» یا Relic است، از روش‌های بسیار کوچک و قابل حمل قرار داد! این تکنولوژی که آن جایی که داده‌های خام حاصل از این کپی‌گرفتن، پیشرفته فشرده‌سازی و ذخیره داده استفاده می‌کند. از آن جایی که داده‌های مشابه فشرده‌سازی متون، عکس و ویدیو عمل باشند، می‌توان در فشرده‌سازی آنها از روش‌های مشابه فشرده‌سازی متون، عکس و ویدیو عمل کرد. می‌خواهیم بررسی کنیم که چنین درجه‌ای از فشرده‌سازی چگونه ممکن است!

گاهی در حین ذخیره‌سازی داده‌ها، حجم ذخیره‌شده بسیار زیاد می‌شود. برای مثال، یک فیلم سینمایی ۱۲۰ دقیقه‌ای را در نظر بگیرید که آهنگ فریم‌های آن ۳۰fps باشد، و ابعاد آن نیز ۷۲۰ در ۱۰۸۰ پیکسل باشد. اگر هر پیکسل را یک عدد ۳ بایتی در نظر بگیریم، آن‌گاه به‌وسیله یک محاسبه ساده نتیجه می‌شود که حدود ۵۰۳ گیگابایت حافظه مورد نیاز خواهد بود. لیکن همان‌طور که می‌دانید این چنین نیست، و دلیل آن فشرده‌سازی‌ای می‌باشد که بر ویدیوهای اعمال می‌شود.

یک الگوریتم ساده برای فشرده‌سازی بدین گونه می‌باشد که داده‌های تکراری را حذف کرده و به جای آنها از یک کد جایگزین استفاده کنیم.

در طی این مسئله، یک استرینگ ورودی به شما داده می‌شود که فقط شامل $a - z$ $A - Z$ و فاصله هستند و شما موظف هستید که کلماتی که بیش از یک بار تکرار شده‌است را کدگذاری کنید. منظور از کلمه، دنباله‌ای از کاراکتر‌هاست که فقط شامل حروف الفباست و در رشته اصلی سر و ته اش فاصله، آغاز، و

انتهای متن است. کدگذاری نیز، به شیوهٔ ساده‌ای اجرا خواهد شد. بدین صورت که از صفر شماره‌گذاری را شروع کرده، و به هر کلمه‌ای که بیش از یکبار تکرار می‌شود عددی جدید تخصیص می‌دهید، سپس این عدد جایگزین کلمه در متن می‌شود. توجه کنید که اگر بیش از یک مرتبه یک کلمه پشت سر هم تکرار شود، در جلوی عدد متناظر کدگذاری، نماد $[n]$ اضافه می‌شود و فقط یکبار آن کلمه نشان‌داده می‌شود.

برای مثال:

hey hey hey

برابر است با

0[3]

وروُدی

یک متن شامل کاراکترهای $[a - z A - Z]$ که طول آن بین ۰ تا ۱۰۰۰۰ می‌باشد به شما ورودی داده می‌شود. توجه کنید که حداکثر طول یک کلمه ۶۳ کاراکتر خواهد بود.

خروجی

نخست دیکشنری‌ای که طبق آن فشرده‌سازی انجام شده‌است و سپس متن فشرده‌شده را چاپ کنید، بلکه بتوانیم در آینده از خودمان کپی دائمی بگیریم!

وروُدی نمونه ۱

hey hey hey

خروجی نمونه ۱

0: hey
0[3]

وروُدی نمونه ۲

hey i am good are you good hey how are you doing so good

خروجی نمونه ۲

0: hey
1: good
2: are
3: you
0 i am 1 2 3 1 0 how 2 3 doing so 1

ورودی نمونه ۳

computer engineering is a dynamic and essential discipline that merges the

خروجی نمونه ۳

0: computer
1: engineering
2: is
3: a
4: and
5: that
6: the
7: of
8: design
9: hardware
10: software
11: engineers
12: in
13: this
14: field
15: physical
16: such
17: as
18: to
19: systems

20: are

0 1 2 3 dynamic 4 essential discipline 5 merges 6 principles 7 electrical 1

ورودی نمونه ۱۴

in the name of Allah i greet you for being accepted into this university an

خروجی نمونه ۱۴

0: in
1: the
2: of
3: i
4: you
5: being
6: accepted
7: into
8: this
9: and
10: to
11: be
12: are
13: a
14: department
15: talk
16: so
17: can
18: here
19: is
20: find
21: do
22: not
23: about
24: interesting

0 1 name 2 Allah 3 greet 4 for 5 6 7 8 university 9[3] 2 course 7 1 hardest

مولفه‌های همبندی (Mikoshi)

- محدودیت زمان: ۰.۲ ثانیه
 - محدودیت حافظه: ۲۵۶ مکابایت
-

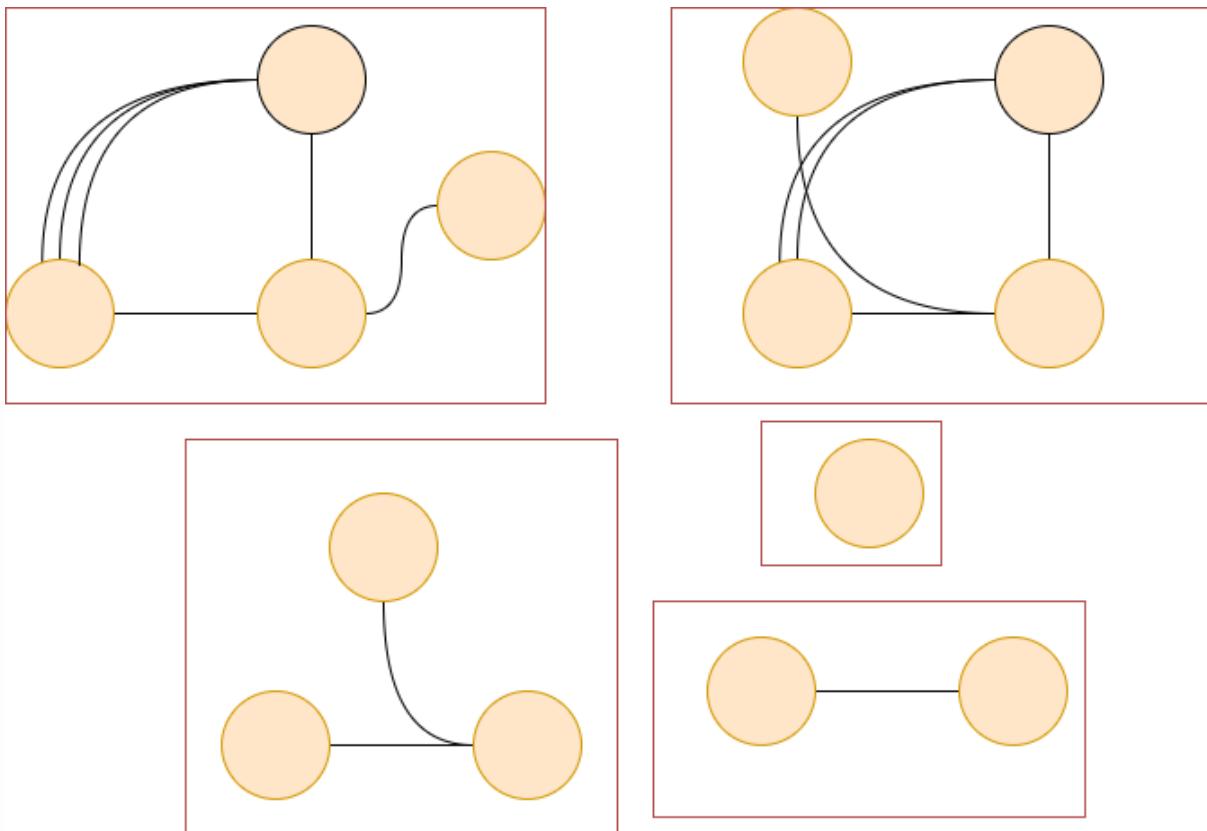




برج آراساکا بلندترین ساختمان در نایت‌سیتی (Night City) است و زیر صد و چهل و اندی طبقه‌اش، یکی از مکان‌های دسترسی به می‌کوشی قرار دارد (Mikoshi - 神輿). در می‌کوشی کپی‌های دیجیتالی از مغازه‌ها آدم ذخیره می‌شوند. بنا به دلایلی که خودتان هم الان به یاد نمی‌آورید، شما تصمیم می‌گیرید که از در اصلی برج آراساکا وارد بشوید و خودتان را به زیرزمین و سپس می‌کوشی برسانید (خدا بهتان رحم کند...). اما از آنجایی که هنوز کاملاً عقل خود را از دست نداده‌اید، تصمیم می‌گیرید یک نقشه از ساختمان برج تهیه کنید و مسیرتان را از قبل در آن مشخص کنید. شما دست به قلم و دست به گد می‌شوید تا همه مسیرها را تحلیل کنید و به یاد می‌آورید که هنوز مقدار کمی نظریه گراف از دبیرستان به خاطر دارید...



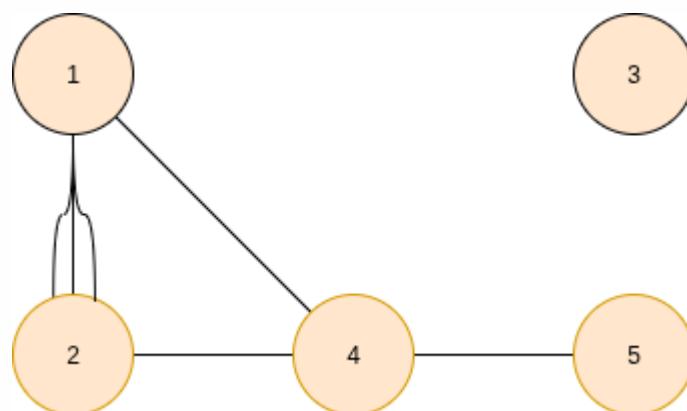
در نظریه گراف یک مولفه همبندی، یک زیرگراف از گراف G است که بین هر دو رأس آن حداقل یک مسیر وجود داشته باشد و هیچ رأس دیگری نباشد که به رئوس این مولفه، مسیری داشته باشد. در واقع، تعداد مولفه‌های همبندی یک گراف برابر با تیکه‌هایی از گراف است که هر تیکه همبند است و بین هیچ دو تیکه‌ای مسیری وجود ندارد. برای مثال مولفه‌های همبندی گراف زیر نشان‌داده شده‌است. که شامل پنج مولفه همبندی می‌باشد.



▼ اطلاعات بیشتر در رابطه با مولفه‌های همبندی گراف
[مرجعی برای دانش بیشتر \(اختیاری\)](#)

در راستای بررسی تعداد مولفه‌های همبندی گراف، شیوه‌های مختلفی برای حل این مسئله وجود دارد، یکی از روش‌ها، استفاده از **ماتریس مجاورت** می‌باشد. در صورتی که نیاز دارید درمورد ماتریس مجاورت بیشتر بدانید، متن پنهان‌شده زیر را بخوانید.

▼ اطلاعات بیشتر در رابطه با ماتریس مجاورت
 ماتریس مجاورت، متناظر گراف G که رئوس آن $V = \{v_1, v_2, v_3, \dots, v_n\}$ می‌باشد، برابر با یک ماتریس $n * n$ است که خانه (j, i) آن برابر با تعداد یال‌های بین دو راس v_i و v_j می‌باشد.



برای مثال گراف فوق، متناظر ماتریس زیر است.

$$\begin{bmatrix} 0 & 3 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

در صورت تمایل به مطالعه بیشتر در این زمینه، می‌توانید [از این لینک](#) استفاده کنید.

ورودی

شما نخست یک عدد n دریافت کرد و در n خط بعدی، در هر خط n عدد دریافت شده، متناظر ماتریس مجاورت شماست.

$$0 \leq n \leq 1000$$

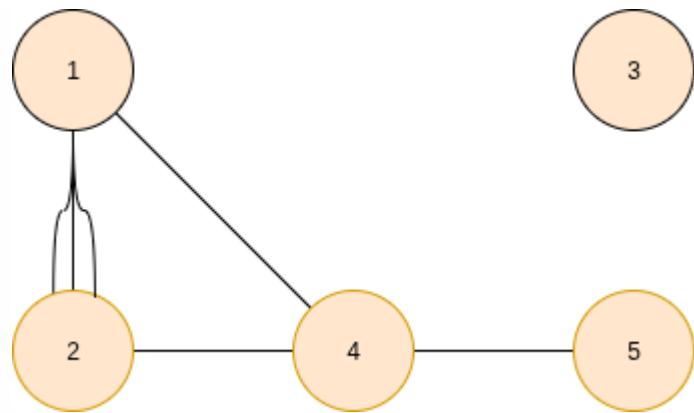
و در راستای تخصیص فضای ذخیره‌سازی لازم به ذکر است که بین هر دو راس حداقل ۵ یال وجود دارد.

خروجی

یک عدد متناظر تعداد مولفه‌های همبندی گراف وارد شده تحت عنوان خروجی چاپ می‌شود.

ورودی نمونه ۱

```
5
0 3 0 1 0
3 0 0 1 0
0 0 0 0 0
1 1 0 0 1
0 0 0 1 0
```



خروجی نمونه ۱

۲

یک مولفه، ۵، ۴، ۲، ۱ و مولفه دیگر

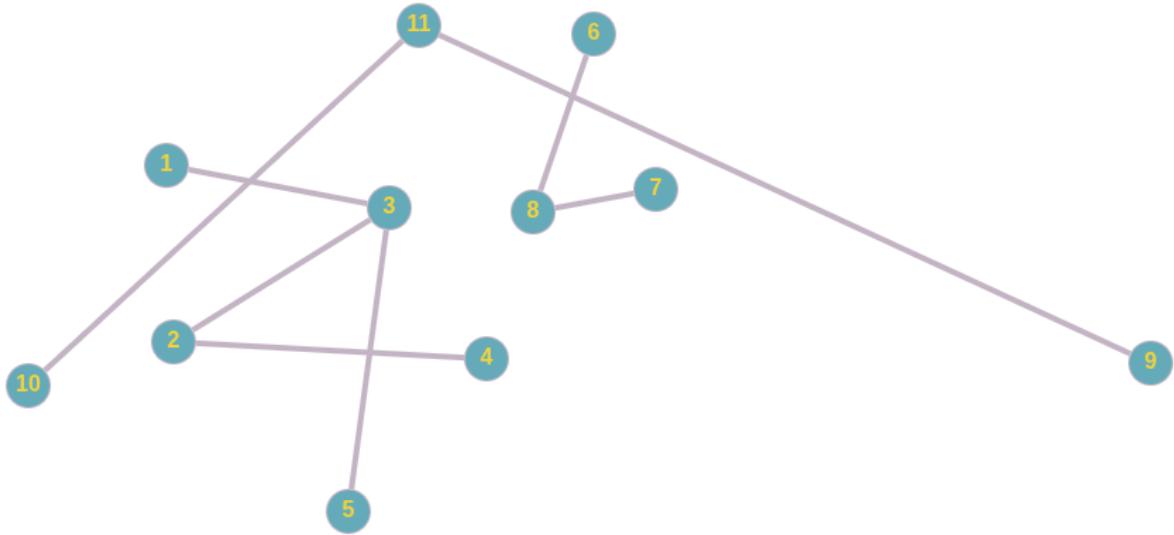
نکته: این تست معادل تست شماره ۱ داوری می‌باشد.

ورودی نمونه ۲

```

11
0 0 1 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0 0
1 1 0 0 1 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0

```



خروجی نمونه ۲

3

که مولفه‌های آن به شکل زیر است:

- 1, 2, 3, 4, 5
- 6, 7, 8
- 9, 10, 11

نکته: این تست معادل تست شماره 9 داوری می‌باشد.

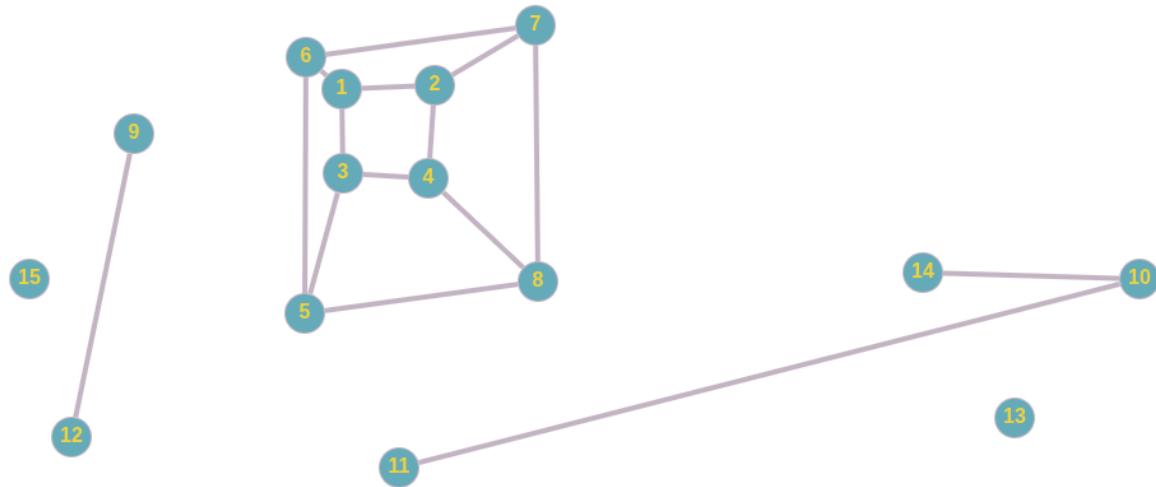
ورودی نمونه ۳

```

15
0 1 1 0 0 1 0 0 0 0 0 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 1 0 1 0 0 0 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 1 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

```

```
0 0 0 0 0 0 0 0 0 1 0 0 1 0  
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```



خروجی نمونه ۳

5

مولفه‌ها به شکل زیر هستند.

- 1, 2, 3, 4, 5, 6, 7, 8
- 9, 12
- 10, 11, 14
- 13
- 15

پردازشگر سامورایی متن (Wakkako's Favorite)

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



بانو واکاکو اوکادا (ژاپنی : 岡田 和歌子 Wakako Okada، انگلیسی: Wakako Okada) به عنوان یک فرد مهم در "تجارت" بسیار سرش شلوغ است و هر روز صدھا پیام کاری دریافت می‌کند. از آنجایی که ایشان حوصله جوابدهی دستی به همهٔ این پیامها را ندارند، البته قبل این هم خیلی حوصله‌دار تشریف نداشتند، از شما کمک می‌خواهند که برایشان یک نرمافزار کمکی برای جواب دادن سریع‌تر به پیام‌هایشان بنویسید. از آنجایی که

شما دوست ندارید توسط شمشیر سامورایی ایشان به دو شقة مساوی تقسیم شوید، به سرعت شروع به کدزنی می‌کنید...

در این سوال، به شما متنی داده می‌شود و باید تعدادی عملیات روی آن انجام دهید. روی این متن یک نشانگر متن (cursor) قرار دارد و به صورتی که در ادامه شرح داده می‌شود، قادر به حرکت است. یک سری از دستورات که داده می‌شود باید با در نظر گرفتن مکان کنونی نشانگر روی متن لحاظ شود. توجه کنید که نشانگر در هر لحظه، روی یکی از کاراکترها قرار دارد و بین آن‌ها قرار نمی‌گیرد. در این سوال منظورمان از کلمه دنباله‌ای متوالی از *z* – *A* – *Z* و *a* – *A* (حروف انگلیسی) است. به طور مثال متن زیر دارای ۴ کلمه می‌باشد:

hello IT5is me

که عبارت اند از hello, IT, is, me از

لیست عملیات‌ها

ریشه‌ای کردن کلمات

یک سری کلمات به عنوان کلمه ریشه‌ای معرفی می‌شوند که در بخش زیر لیست آن‌ها قابل مشاهده می‌باشد:

▼ لیست کلمات ریشه‌ای

go

run

play

right

light

happy

quick

art

rat

star

دستور:

rooting

با وارد شدن دستور بالا شما باید تک تک کلمات متن را به صورت زیر بررسی کنید:

اگر یکی از کلمات ریشه‌ای، زیر کلمه‌ای از آن کلمه باشد، به جای کلمه درون متن، آن کلمه ریشه‌ای را قرار می‌دهیم. منظور از زیرکلمه آن است که با حذف تعدادی حرف از ابتدا و/یا انتهای آن، معادل آن کلمه ریشه‌ای ایجاد بشود.

به طور مثال اگر متن ما این باشد:

I was frightened to disagree with him.

متن ما، به متن زیر تبدیل می‌شود:

I was right to disagree with him.

که در واقع، همان‌طور که مشاهده می‌کنید right frightened با right جایگزین شده‌است. (با حذف fr از ابتدای کلمه و ened از انتهای آن، right frightened به right تبدیل می‌شود). (اگر کلمه‌ای چند زیر کلمه از کلمات ریشه‌ای داشت زیر کلمه‌ای را در نظر بگیرید که از بقیه بالاتر در لیست کلمات ریشه‌ای آمده است).

توجه کنید که اگر پس از این عملیات عدد مربوط به کرسر بزرگتر از طول رشته شد، کرسر را روی آخرین کاراکتر رشته قرار دهید.

مکان نشانگر

location

با وارد شدن دستور بالا، باید مکان کنونی نشانگر را در یک خط جدا نمایش دهید. در نمایش مکان نشانگر، باید اندیس کاراکتری که نشانگر روی آن قرار دارد را چاپ کنید. برای مثال فرض کنید نشانگر روی کاراکتر

هشتم قرار دارد. این مکان را مانند زیر نمایش می‌دهیم:

7

مکان اولیه نشانگر روی اولین کاراکتر است. به عبارتی دیگر در اول کار با دادن این دستور، 0 چاپ خواهد شد.

حرکت‌دادن نشانگر

move <amount>

با این دستور، باید نشانگر را به میزان گفته‌شده جابه‌جا کنید. در صورتی که عدد داده شده مثبت باشد، نشانگر را به همان تعداد کاراکتر روی متن به راست جابه‌جا کنید و در صورتی که عدد وارد شده منفی باشد، نشانگر را به چپ حرکت دهید.

در صورت خارج شدن نشانگر از متن، دستور اجرا نمی‌شود و عبارت زیر در یک خط جدا چاپ می‌شود:

Not enough characters.

اضافه‌کردن کلمه

add <word>

در این دستور سمت راست کلمه‌ای کرسر روی آن قرار گرفته است، کلمه مشخص شده در دستور را اضافه می‌کنیم. اگر کرسر روی کاراکتر یک کلمه بود، کنار آن کلمه اسپیس می‌گذاریم و بعد کلمه جدید را اضافه می‌کنیم. اگر کرسر روی کاراکتری بود که جز کلمه‌ای نیست، بلافاصله کنار آن کاراکتر کلمه را اضافه می‌کنیم. اگر کرسر روی کاراکتر نگاه کاراکتر یک کلمه بود یک اسپیس نیز اضافه می‌بعد از نوشتن این کلمه نگاه کاراکتر بعدی می‌کنیم؛ اگر کاراکتر یک کلمه بود یک اسپیس نیز اضافه می‌کنیم. به *مثال*های زیر برای درک بهتر این دستور توجه کنید:

وروودی

FOP is really good.

فرض کنید کرسر روی کاراکتر A باشد. اگر دستور به صورت:

add very

خروجی به صورت زیر خواهد بود:

FOP is very really good.

اگر کرسر روی کاراکتر نقطه بود:

FOP is really good.very

اگر کرسر روی اولین کاراکتر از جمله زیر باشد و همان دستور بالا را داشته باشیم:

we,they

متن تبدیل می‌شود به:

we very,they

اگر کرسر روی کاراکتر سوم(اسپیس) از جمله زیر باشد و همان دستور بالا را داشته باشیم:

we ran

متن تبدیل می‌شود به:

we very ran

و در واقع بلافاصله پس از کلمه‌ای که روی آن داریم اسپیس گذاشته و سپس کلمه را می‌نویسیم و باقی متن بدون تغییر بعد از آن می‌آید.

برعکس کردن کلمه

reverse

در این دستور، شما حروف کلمه‌ای را که کرسر روی آن قرار دارد برعکس می‌کنید.

به طور مثال، اگر کرسر روی کلمه clean باشد پس از این دستور به جای این کلمه، کلمه naelc قرار داده می‌شود. (clean با naelc جایگزین می‌شود).

*توجه: اگر کرسر روی حرف a بوده باشد در متن جدید روی حرف a قرار خواهد بود.

*نکته: اگر کرسر روی کلمه‌ای قرار نداشت (روی اسپیس یا علائم نگارشی باشد) نیازی نیست تغییری در متن ایجاد کنید.

تعجب

WOW

در این دستور شما دقیقا سمت راست کلماتی که ویژگی‌های زیر را دارند علامت تعجب اضافه می‌کنید :

۱. تعداد حروف کلمه حداقل ۵ باشد.
۲. کلمه حداقل شامل دو حرف صدادار می‌باشد.

به طور مثال، جمله زیر:

The sky is so majestic, so I love it.

به جمله زیر تبدیل می‌شود:

The sky is so majestic!, so I love it.

*نکته: حروف i, o, u, e, a صدادار هستند.

حروف بزرگ یا کوچک

<upper | lower>

در این دستور، کلمه‌ای که کرسر روی آن است، از کاراکتر کرسر تا انتهای کلمه، عملیات upper (تبدیل کاراکترهای متن به حرف بزرگ در صورت امکان) یا lower را (تبدیل کاراکترهای متن به حرف کوچک در صورت امکان) اجرا می‌کنیم.

توجه کنید که سه دستور بالا تنها روی کاراکترهای *z* – *A* – *Z* و *a* – *A* – *Z* (حروف انگلیسی) تاثیر می‌گذارند و در غیر این صورت تاثیری روی کاراکتر اعمال نمی‌شود.

به طور مثال، اگر متن زیر را داشته باشیم و کرسر روی کاراکتر دوم متن باشد:

FOP very is really good.

متن ما با دستور lower به متن زیر تبدیل می‌شود:

Fop very is really good.

تغییر حروف

change <amount>

مکان کنونی نشانگر را X می‌نامیم. عملیات change را در بازه X + amount تا X + amount اجرا می‌کنیم.

*توجه: عملیات روی جایگاه X اجرا می‌شود، اما روی جایگاه X + amount اجرا نمی‌شود.

عملیات change به این صورت است، که هر حرف با حرف بعدی اش در الفبا عوض می‌شود. به طور مثال کاراکترهای f را با کاراکتر g و تمامی کاراکترهای F را با کاراکتر G عوض می‌شود.

*توجه: z با کاراکتر a و تمامی کاراکترهای Z با کاراکتر A عوض می‌شود.

مکان نشانگر نیز به تعداد amount جلو می‌رود. اگر تعداد کاراکترهای موجود بعد از مکان کنونی نشانگر از amount کمتر باشد و نشانگر پس از طی کردن amount کاراکتر از متن خارج شود، دستور اجرا نمی‌شود و عبارت زیر در یک خط جدا چاپ می‌شود:

Not enough characters.

مقدار amount در دستور بالا عددی نامنفی است. به ازای amount برابر صفر، دستور بالا کاری انجام نمی‌دهد.

به طور مثال، اگر متن زیر را داشته باشیم و کرس روى کاراکتر اول متن باشد و دستور change 3 را بدھیم:

i am here.

متنمان به متن زیر تبدیل خواهد شد:

j bm here.

و کرس در متن جدید روی کاراکتر m قرار می‌گیرد.

ورودی

در خط اول ورودی، متن اولیه به شما داده می‌شود. در خطهای بعدی هم دستورات به ترتیب در ورودی نوشته می‌شوند تا در نهایت به ## بررسیم و برنامه به پایان برسد. طول رشته ورودی کمتر از ۳۰۰ کاراکتر است. همچنین تضمین می‌شود در طول کل برنامه طول رشته کمتر از ۵۰۰ کاراکتر باشد.

خروجی

در حین اجرای برنامه، خروجی‌هایی که در بعضی دستورات ذکر شده است را چاپ کنید. در انتهای برنامه نیز متن نهایی را در خروجی چاپ کنید. فراموش نکنید که شمشیر سامورایی بانو اوکادا همچنان به شما چشمک می‌زند...

ورودی نمونه ۱

If you are going through hell, keep going.

add haha

move 100

reverse

move 4

```
upper  
change 5  
location  
WOW  
###
```

خروجی نمونه ۱

```
Not enough characters.  
9  
fI hBIB zou are going! through! hell, keep going!.
```

ورودی نمونه ۲

```
He finished university quickly and graduated.  
move 3  
WOW  
add his  
rooting  
move -2  
location  
lower  
###
```

خروجی نمونه ۲

```
1  
He finished his! university! quick! and graduated!.
```

ورودی نمونه ۳

```
Hello World  
move 6  
upper  
add amazing  
move -7  
reverse
```

```
WOW  
change 5  
location  
###
```

خروجی نمونه ۳

```
Not enough characters.  
11  
Hello! EMSPW amazing!
```

ورودی نمونه ۴

```
abcZ y!  
move 3  
change 2  
move 2  
lower  
add GREAT  
move -4  
reverse  
location  
###
```

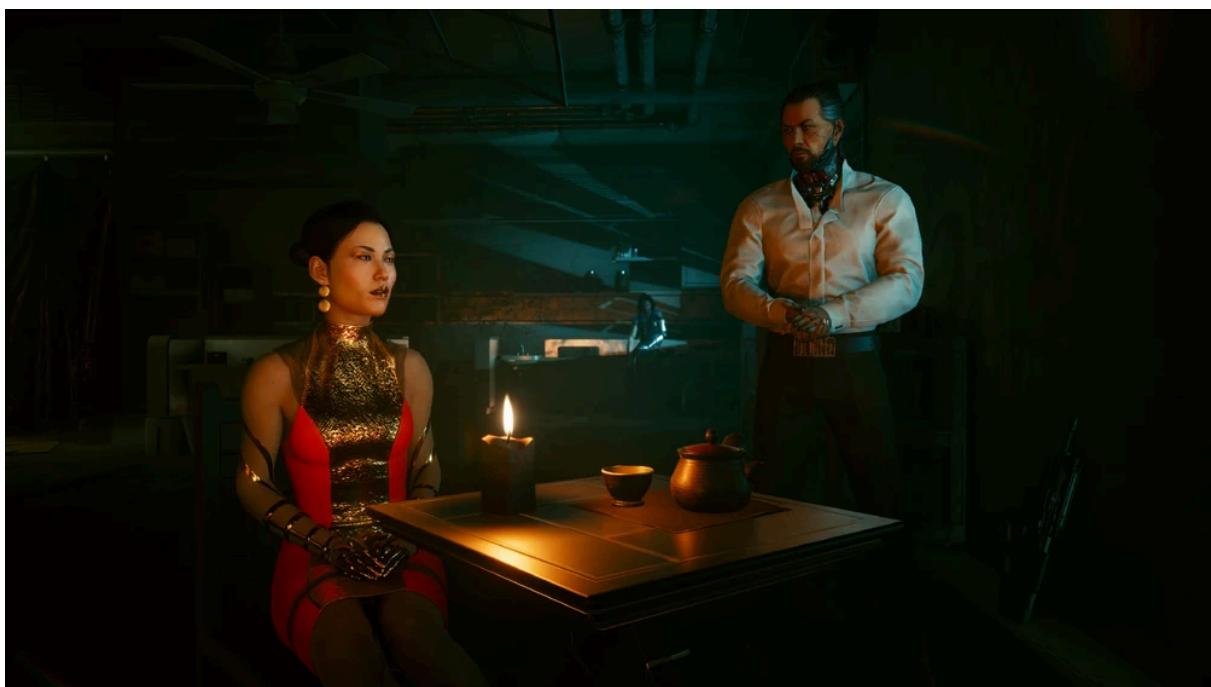
خروجی نمونه ۴

```
Not enough characters.  
1  
Acba y GREAT!
```

پالیندروم (Hanako & Takemura)

• محدودیت زمان: ۲ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت



جناب گورو تاکمورا (ژاپنی: 五郎 村竹، انگلیسی: Goro Takemura) بعد از تعدادی اتفاق غیرمتربقه، به ناچار تصمیم می‌گیرد که دختر صاحب کارش، هاناکو آراساکا (ژاپنی: 荒坂 華子، انگلیسی: Hanako Arasaka) را به منزلش بیاورد تا از او حفاظت کند. همان‌طور که می‌توانید حدس بزنید، بانو هاناکو اصلاً از این اتفاق خوشحال نیستند و حتی لب به چای هم نمی‌زنند. جناب تاکمورا از شما عاجزانه کمک می‌خواهد تا فکری به حال این اوضاع بفرمایید! شما پیشنهاد می‌کنید که یک بازی دوستانه برای گذراندن زمان انجام بدهند. هر دو موافقت می‌کنند، اما جناب تاکمورا دم گوش شما می‌گویند که یک روشی پیدا کنید که ایشان بتوانند قبل از اتمام بازی، نتیجه‌اش را بدانند. (طبعاً بانو هاناکو با باخت در بازی شدیداً عصبی خواهند شد!) شما فوراً دست به کار می‌شوید.

در این سوال، دو بازیکن به نوبت روی یک رشته در حال بازی کردن هستند. در هر نوبت یک بازیکن می‌تواند یکی از حروف اول یا آخر رشته را حذف کند. برنده کسی است که بتواند در نوبت خودش رشته را به حالت پالیندروم برساند.

▼ بیشتر بخوانید (پالیندروم)

پالیندروم (Palindrome) یعنی کلمه، عدد، جمله یا رشته‌ای از حروف که اگر آن را از ابتداء تا انتهایا بازیکن level, abba, nan, mom, salamalas, A, SAFIR-RIFAS ابتدابخوانی، یکسان باشد. مثال:

ورودی

ورودی خط نخست، شامل یک عدد طبیعی t که نشاندهنده تعداد رشته‌هایی است که روی آنها بازی می‌شود. در هر کدام از یک t خط بعدی، یک رشته داده می‌شود و شما باید استنتاج کنید که اگر هر بازیکن به صورت عاقلانه و با بهترین استراتژی بازی کند، کدام یک برنده می‌شود. (یا به عبارتی کدام بازیکن استراتژی برد دارد!)

▼ بیشتر بخوانید (استراتژی برد)

اگر وضعیت بازی را با حالت‌ها (States) نشان دهیم:

- حالت‌هایی وجود دارند که بازیکن جاری می‌تواند از آنها برد خود را تضمین کند → اینها Winning Positions هستند.
- در این حالت‌ها، استراتژی بُرد مجموعه‌ای از حرکات است که همیشه بازی را از حالت فعلی به حالتی می‌برد که برای حریف باخت است.

$$1 \leq t \leq 500$$

تضمین می‌شود طول هر رشته حداقل 500 باشد. همچنین رشته‌ها تنها شامل کarakترهای ASCII هستند.

خروجی

در t خط خروجی، برندهٔ هر تست را مشخص کنید. اگر نفر اول برنده می‌شود First را چاپ کنید و اگر نفر دوم برنده می‌شود Second.

مثال

ورودی نمونه ۱

3
one
all
alli

خروجی نمونه ۱

Second
First
Second

توضیح:

برای رشته نخست، اگر بازیکن اول هر حرکتی انجام بدهد. بازیکن دوم با حذف یکی از حروف دیگر به رشته تکحرفی میرسد که تک رشته پالیندروم است!

برای رشته دوم، بازیکن اول با حذف حرف a، رشته را پالیندروم میکند.

برای رشته سوم، بازیکن اول اگر a را حذف کند، بازیکن دوم با حذف ا رشته را پالیندروم میکند و اگر بازیکن اول ا را حذف کند، بازیکن دوم با حذف a رشته را پالیندروم میکند.

ورودی نمونه ۲

2
seyed
Amoooooo

خروجی نمونه ۲

Second
Second

توضیح: برای رشته نخست، بازیکن اول اگر s را حذف کند، بازیکن دوم با حذف d رشته را پالیندروم میکند و اگر بازیکن اول d را حذف کند، بازیکن دوم با حذف s رشته را پالیندروم میکند.

درمورد رشته دوم، اثبات آن که نفر دوم استراتژی برد دارد به خواننده واگذار می‌گردد. اثبات این مهم، با
حالتبندی مقدور است.

وروودی نمونه ۳

1
abba

خروجی نمونه ۳

First

وروودی نمونه ۴

1
ab

خروجی نمونه ۴

First