

شاد باشید

یک مؤسسه‌ی فرهنگی به دلیل شرایط جامعه و لزوم وجود شادی میان مردم، قصد دارد برنامه‌های شادی بخش تهیه و پخش کند، اما در تعیین میزان شادی یا ناراحتی مردم به مشکل خورده است. این مؤسسه از شما می‌خواهد با دریافت حالت یک نفر، میزان شادی و ناراحتی فرد ورودی را خروجی دهید تا از این اطلاعات برای برنامه‌ریزی استفاده کند.

ورودی

در یک خط از ورودی، یک رشته با طول بزرگ‌تر از ۲ حاوی تنها یک کاراکتر : در ابتدا و چند کاراکتر (یا) در ادامه‌ی رشته وارد می‌شود.

خروجی

برایند حالت یک فرد را با پیشوند so و پسوند happy یا sad در خروجی نمایش دهید.

در صورت تساوی درجه‌ی خوشحالی و ناراحتی یک فرد، | : را چاپ کنید.

هر (یک درجه به ناراحتی فرد می‌افزاید.

هر) یک درجه به خوشحالی فرد می‌افزاید.

مثال

ورودی نمونه ۱

(:)))

خروجی نمونه ۱

so so happy

درجه خوشحالی و ناراحتی فرد به واسطه وجود 4 تا (و 2 تا) به ترتیب 4 و 2 است. براینده حالت روحی فرد 2+ و فرد happy است . پس باید ۲ پیشوند so برای نشان دادن میزان شادی قبل از happy چاپ شود.

ورودی نمونه ۲

:((((

خروجی نمونه ۲

so so so so so sad

ورودی نمونه ۳

:)(

خروجی نمونه ۳

:|

ماشین حساب رشته‌ای

به دنبال نبود ماشین حسابی برای رشته‌ها، از شما خواسته شده چنین چیزی طراحی کنید تا خلأ نبود این برنامه‌ی مهم پر شود.

ماشین حساب رشته‌ای ما از چندین عملیات مهم پشتیبانی می‌کند:

۱. `to_upper` : این دستور تمام کاراکترهای کوچک یک رشته را بزرگ می‌کند. مثال:

input:
`to_upper i am mahmood.`

output:
`I AM MAHMOOD.`

۲. `to_lower` : برعکس دستور قبل، تمام کاراکترهای بزرگ را کوچک می‌کند. مثال:

input:
`to_lower i aM MahmooD.`

output:
`i am mahmood.`

۳. `camel_case` : شبیه فرمت `camelCase` ، یک جمله را به این شکل در می‌آوریم. چند قانون بین خود می‌گذاریم:

- کاراکتر اول هر کلمه‌ای که قبلش کاراکتری غیر از کاراکترهای الفبای لاتین بود، باید بزرگ شود.
- سایر کاراکترهای هر کلمه (غیر از اولین کاراکتر) باید کوچک شوند.
- هر کاراکتری غیر از اعداد و الفبای انگلیسی باید حذف شود.
- کاراکتر اول جمله باید کوچک باشد.

مثال:

input:
`camel_case Beautiful clouds in sky, will rain soon. at5aM`

output:
beautifulCloudsInSkyWillRainSoonAt5Am

برای مطالعه بیشتر درباره‌ی camelCase ، در اینترنت جست‌وجو کنید.

نکته: خیلی خوب است که از این قاعده برای نام‌گذاری متغیرهای جاوا استفاده کنید.

۴. capital : کاراکتر اول تمام کلمات در صورتی که کوچک بود، بزرگ شود. مثال:

input:
capital beautiful clouds in sky, will rain soon.

output:
Beautiful Clouds In Sky, Will Rain Soon.

نکته: برای تشخیص هر کلمه به این نکته توجه کنید که بین کلمات کاراکترهای عددی یا کاراکترهای غیرانگلیسی می‌آید.

۵. encode : این دستور یک رشته را مطابق الگوی رمزگذاری ما رمزگذاری می‌کند.

الگوی رمزگذاری ما:

- ابتدا رشته برعکس می‌شود.
- سپس کاراکترهایی که ایندکس زوج دارند به ترتیب در کنار هم و کاراکترهای با ایندکس فرد در کنار هم قرار می‌گیرند (ایندکس‌ها از 0 شروع می‌شوند).
- سپس این دو گروه را با استفاده از جدا کننده #-# در کنار هم قرار می‌گیرند (تضمین می‌شود در جمله وارد شده عبارت #-# وجود نداشته باشد).

مثال:

input:
encode Beautiful Clouds.

output:
.doClftaB#-#sul uiue

۶. decode : یک جمله‌ی رمزگذاری‌شده مطابق الگوی رمزگذاری ما را رمزشکنی می‌کند (تضمین می‌شود آخر جمله وارد شده *space* نداریم). مثال:

input:

decode .doClftaB#-#sul uiue

output:

Beautiful Clouds.

۷. 0FF : با این دستور، اجرای برنامه به پایان می‌رسد.

تضمین می‌شود دو طرف جمله وارده اسپیس نداشته باشیم.

بازی موازی

از وقتی که دانشگاه تعطیل شده سنا مشغول گشت‌وگذار در اینترنت شده و به دنبال یادگیری است. او برای تفریح URL هایی را که وارد می کند در دفترش ثبت می‌کند و پیش خود نگه می‌دارد. سنا روزی پیش دخترخاله‌ی خود کیمیا می‌رود تا بازی‌ای انجام دهد. آن‌ها URL هایی را که سنا نوشته است را با الگوی خود تطبیق می‌دهند و اگر مطابق بود، قسمت *domain* آن را برعکس می‌کنند و می‌نویسند.

از آن‌جایی که نوشته های سنا زیاد است، از شما کمک می‌خواهد.

در باره‌ی قسمت های مختلف URL در اینترنت جست‌وجو کنید.

در سؤال، بخش‌های *domain* ، *subdomain* و *TLD* مطرحند:



URL های الگوی سنا و کیمیا از این قرار است:

- قسمت *subdomain* آن شامل سه حرف انگلیسی باشد.
- قسمت *domain* آن شامل حروف ، اعداد و _ بوده و حداقل شامل ۲ کاراکتر باشد.
- قسمت *TLD* یکی از مقادیر `net` , `com` , `ir` , `org` باشد.

مثالی از چند URL معتبر:

www.saljfnfnke.org
abc.google.com
new.abc_12.ir

ورودی

در تعداد نامشخصی خط از ورودی، در هر خط یک *URL* وارد می‌شود.

خروجی

اگر *URL* با الگو مطابقت داشت، برعکس *domain* آن و اگر نبود، `This URL is uncorrect!` را به ازای هر *URL* چاپ کنید.

ورودی نمونه

abc.google.com
new.abc_12.ir
www.abc-12.ir
irr.W3Schools.net

خروجی نمونه

abc.elgoog.com
new.21_cba.ir
`This URL is uncorrect!`
irr.sloohcS3W.net

دکتر بدخط

بهنام که از شدت فشردگی ترم دچار بیماری شده، پیش دکتر خود می‌رود تا او را معاینه کند. دکتر به او نسخه‌ای می‌دهد که شامل قرص‌هایی است که حال بهنام را خوب کند. وقتی که بهنام به داروخانه می‌رود، نسخه را به متصدی داروخانه می‌دهد تا داروهایش را بدهد. متصدی که خط دکتر را نمی‌تواند بخواند، از شما کمک می‌خواهد که برایش ترجمه کنید.

داروها به شرح زیر هستند:

- قرص آسپرین: شامل حروف انگلیسی و اعداد (هر دو مورد هستند) که به وسیله‌ی _ از یکدیگر جدا شده‌اند.
- قرص ژلوفن: شامل حروف انگلیسی، اعداد، _ ، - و . (الزامی) است.
- قرص ویتامین: شامل حروف انگلیسی، اعداد، _ ، - ، . و پرانتز (الزامی) است.

نکات

- در قرص ژلوفن نباید _ و - به . چسبیده باشند.
- در قرص ویتامین پرانتز ها باید یکدیگر را خنثی کنند (به ازای هر) یک (باشد) و اگر به ازای هر) یک (نبود یا این‌گونه) قرار داشتند صحیح نیست.
- اگر شروط بالا رعایت نشده بودند، عبارت Tamom kardim! چاپ شود.

چند مثال معتبر:

```
abc_123
12_ddf
abc.00fd
frg(df)df
```

ورودی

در تعداد نامشخصی خط از ورودی، در هر خط نام یک قرص وارد می‌شود.

خروجی

به ازای هر قرص اگر نام قرص با یکی از شرایط ذکرشده مطابقت داشت، نام قرص و اگر نداشت، عبارت Tamom kardim! را چاپ کنید.

ورودی نمونه

```
abc_123
abc.00fd
(df)(fdd
frg(df)df
)dcdc(dc
12_ddf
```

خروجی نمونه

```
Aspirin
Gelofen
Tamom kardim!
Vitamin
Tamom kardim!
Aspirin
```

سیاس پرحاشیه (امتیازی)

استفاده از کالکشن در این تمرین مجاز نیست.

همان‌طور که می‌دانید، برای زیباسازی شکل و شمایل تگ‌های *HTML* از *CSS* استفاده می‌شود.

برای استفاده *CSS* در *HTML* سه راه وجود دارد:

۱. نوشتن کل فایل *CSS* و شناساندن آن فایل به فایل *HTML*:

```
Tagname/Id/class/... {  
  Css styles...  
}
```

مثال:

```
1 | #link1 {  
2 |   margin: 10px;  
3 |   color: red;  
4 | }  
5 | #header_2 {  
6 |   font-family: BTitr;  
7 | }
```

۲. نوشتن کد *CSS* در تگ استایل (`<style> ... </style>`)

۳. استفاده از اتریبیوت `style` در هر تگ

مثال:

```
1 | <a id= "link1" href="google.com" style="color: red; margin: 10px">link to goo
```

حال، محمود می‌خواهد سر و سامانی به کدهای از قبل نوشته شده‌ی خود بدهد و استایل‌های *CSS* را از حالت سوم به حالت اول تبدیل کند.

از آنجایی که ترم اول مجازی بوده و او متأسفانه سر کلاس کارگاه کامپیوتر تقلب کرده، سر رشته‌ای از *HTML* و *CSS* ندارد. او از شما می‌خواهد این کار را با جاوا برایش انجام دهید.

ورودی

یک رشته حاوی کد *HTML* وارد می‌شود که با استفاده از اتریبیوت `style`، به آن ظاهر داده شده است.

خروجی

یک رشته که محتویات فایل *CSS* ما را نشان می‌دهد، با فرمت زیر چاپ کنید:

```
1  #id[space]{[\n]
2  style;[\n]
3  style;[\n]
4  ...
5  }[\n]
6  #id[space]{[\n]
7  style;[\n]
8  style;[\n]
9  ...
10 }
```

مثال

ورودی نمونه ۱

```
1  <html lang="en">
2  <head>
3  <meta charset="UTF-8">
4  <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <title>Document</title>
6  </head>
7  <body>
8  <a href="google.com" style="margin: 10px; color: red;" id = "link1">link to g
9  <div>
10 <form>
```

```

11 <button id="btn" style="color: brown; font-size: large;"/>
12 <p id="header_2" style="font-family: BTitr;">
13 lorem ipsum
14 </p>
15 </form>
16 </div>
17 </body>
18 </html>

```

خروجی نمونه ۱

```

1 #link1 {
2 margin: 10px;
3 color: red;
4 }
5 #btn {
6 color: brown;
7 font-size: large;
8 }
9 #header_2 {
10 font-family: BTitr;
11 }

```

توضیح خروجی نمونه ۱:

در خط ۸، تگ `a` را داریم. اتریبیوت استایل این تگ برابر `style="margin: 10px; color: red;"` است. تگ `a` دیگری وجود ندارد. پس استایل این تگ `margin` و `color` خواهد بود. از خط ۱۱ و ۱۲ نیز استایل مربوط به تگ های `button` و `p` را استخراج می‌کنیم.

نکات

- همه‌ی تگ‌ها از فرمت `<tagname attributes>something</tagname>` یا `<tagname attributes/>` پیروی می‌کنند، یعنی دیگر تگ بدون پایان مثل تگ `
` نخواهیم داشت.
- بین هر مؤلفه می‌توان به تعداد نامحدود اسپیس وارد کرد، مثلاً `id = "tag_id"` هم‌ارز با `id = "tag_id"` است.

- تنها استایل تگ‌هایی را استخراج کنید که همزمان دو اتریبیوت `id` و `style` را دارا باشند.
- ترتیب استایل‌های خروجی از بالا به پایین کد ورودی خواهد بود، یعنی استایل تگی که در ورودی، بالاتر از تگ دیگر باشد، در خروجی نیز بالاتر خواهد بود.
- ترتیب استایل‌های مربوط به هر آیدی از چپ به راست است، یعنی مثلاً اگر در ورودی داشته باشیم `style = "k1: v1; k2: v2;"`، باید در خروجی `k1: v1; k2: v2;` را بالاتر از `k2: v2;` ببینیم.

تضمین می‌شود

- نیازی به هندل کردن معتبر بودن یا نبودن کدهای `CSS` نیست.
- فرمت استایل به صورت `style="key: value; key: value;"` خواهد بود. (از نظر محتوای داخل دابل کوتیشن‌ها)
- تعداد تگ‌هایی که نیاز به استخراج استایل آن‌هاست از ۱۰ مورد تجاوز نمی‌کند.
- تعداد اتریبیوت استایل هر تگ نهایتاً ۱۰ تا است.