روز اول سال

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

با استفاده از دستور $switch$ و توابع کلاس $Math$ برنامه‌ای بنویسید که یک سال را از ورودی گرفته و نام اولین روز آن سال را چاپ کند. برای محاسبهٔ این روز می‌توانید از اطلاعات زیر استفاده کنید:

```
day_code=[year+[(year-1)/4]-[(year-1)/100]+[(year-1)/400]]mod 7
```

جدول نمونه

day	Sun	Mon	Tue	Wed	Thu	Fri	Sat
day_code	0	1	2	3	4	5	6

ورودی

در تنها خط ورودی سوال، به شما یک سال به صورت `YYYY` داده می‌شود.

خروجی

در تنها خط خروجی، نام اولین روز این سال را چاپ کنید.

مثال

ورودی نمونه ۱

```
1978
```

خروجی نمونه ۱

```
Sunday
```

ورودی نمونه ۲

```
2021
```

خروجی نمونه ۲

```
Friday
```

کد ملی

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

سینا در هنگام ثبت‌نام در سامانهٔ واکسیناسیون و در حین وارد کردن کد ملی‌اش، به‌طور اتفاقی یک رقم آن را اشتباه وارد کرده و متوجه شد که سامانه به او خطای نادرست بودن کد ملی را می‌دهد. به همین دلیل تعجب کرد که چطور بدون کلیک کردن روی دکمهٔ استعلام و ارسال کد ملی به سرور مرکزی و جستجو در پایگاه داده، نادرست بودن کد وارده به این سرعت تشخیص داده شده است؟

به همین دلیل به یار همیشگی خود، Google، مراجعه کرده و با جستجو در آن متوجه شد که سامانه قبل از ارسال اطلاعات فرد به سرور، از یک الگوریتم تشخیص صحت کد ملی استفاده می کند. اکنون به سینا کمک کنید که این الگوریتم را با زبان جاوا پیاده‌سازی کند. برای این کار برنامه‌ای بنویسید که کد ملی فرد را گرفته و درست یا نادرست بودن آن را به‌عنوان خروجی به کاربر اطلاع دهد.

کد ملی شماره‌ای 10 رقمی است که از سمت چپ، سه رقم اول کد شهرستان محل صدور شناسنامه ، شش رقم بعدی کد منحصر به فرد برای فرد دارنده شناسنامه را تشکیل می‌دهند. رقم آخر نیز یک رقم کنترل است که از روی ۹ رقم قبلی بدست می‌آید (دقت شود که در سیستم کد ملی معمولا قبل از کد تعدادی صفر وجود دارد (رقم اول و دوم سمت چپ کد ملی ممکن است صفر باشند) و در بسیاری از موارد ممکن است کاربر این صفرها را وارد نکند).

ساختار کد ملی										
ارقام کد	رقم کنترل	9 رقم سمت چپ کد ملی								
موقعیت	1	2	3	4	5	6	7	8	9	10

الگوریتم تشخیص صحت کد ملی:

1- برای محاسبه رقم کنترل از روی سایر ارقام، هر رقم را در موقعیت آن ضرب کرده و حاصل را با هم جمع می‌کنیم (فقط 9 رقم بدون رقم کنترل).

2- اگر باقی‌مانده مجموع بدست آمده به 11 کمتر از 2 باشد، رقم کنترل باید برابر این باقی‌مانده باشد و در غیر این صورت رقم کنترل باید برابر 11 منهای آن قرار داده شود.

3- اگر رقم کنترل کد ملی داده شده با رقم محاسبه شده در مرحله دوم یکسان باشد در این صورت کد ملی درست است و در غیر این صورت کد ملی داده شده نادرست می‌باشد.

ورودی

در تنها خط ورودی سوال، به شما یک کد ملی داده می‌شود.

خروجی

در تنها خط خروجی یکی از کلمات `Valid` یا `Invalid` را بسته به ورودی برنامه چاپ کنید.

مثال

ورودی نمونه ۱

```
2360557891
```

خروجی نمونه ۱

```
Invalid
```

ورودی نمونه ۲

```
1360557891
```

خروجی نمونه ۲

```
Valid
```

ورودی نمونه ۳

```
63522454
```

خروجی نمونه ۳

```
Valid
```

ملالغت

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

سینا که با مشکل خرابی باتری لپ‌تاپ خود مواجه شده است، برای تایپ به موقع تمارین بی‌پی از کامپیوتر قدیمی خود استفاده کرده است. به دلیل قدیمی بودن کیبورد این کامپیوتر، در هنگام تایپ برخی کلمات، اسپیس‌گذاری صحیح در متن رعایت نشده است و برخی از حروف به اشتباه با کاراکترهای کوچک یا بزرگ نوشته شده‌اند.

سینا که ملالغتی را پیشهٔ زندگی خود کرده است و بر درست‌نویسی متن بسیار اصرار دارد، از شما می‌خواهد برنامه‌ای بنویسید که متن تایپ شده توسط این کیبورد قدیمی را با قوانین زیر، ویرایش کند:

۱. بین هر دو کلمه باید دقیقا یک اسپیس قرار گیرد.
۲. هر علامت نگارشی (نقطه، ویرگول، نقطه ویرگول، دو نقطه، علامت تعجب و علامت سوال) باید دقیقا به کلمهٔ قبل خود بچسبد و از کلمهٔ بعد خود دقیقا یک اسپیس فاصله داشته باشد.
۳. تمام متن باید از حروف کوچک تشکیل شود و صرفا کلمات اول جمله (اولین کلمهٔ کل متن و هر کلمه‌ای که بعد از یکی از علائم نگارشی نقطه، نقطه ویرگول، دو نقطه، علامت تعجب و علامت سوال بیاید) باید با حرف بزرگ شروع شوند.

ورودی

در ورودی یک رشته در یک خط با حداکثر طول 2000 کاراکتر از کاربر دریافت می‌شود.

خروجی

در خروجی باید متن ویرایش شده مطابق انتظارات مسئله چاپ شود.

مثال

ورودی نمونه ۱

```
sup !yOU doing Good?
```

خروجی نمونه ۱

```
Sup! You doing good?
```

ورودی نمونه ۲

```
hello .   THIS is  The 6Th bP  exeRcIse ; eZ !
```

خروجی نمونه ۲

```
Hello. This is the 6th bp exercise; Ez!
```

کدگذاری

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

توجه: استفاده از تابع آماده *replaceAll()* و توابع مشابه آن برای حل این سوال غیرمجاز است.

یک مشکل اصلی در ارسال و دریافت پیام‌های الکترونیکی، امکان وجود نویز در هنگام مخابرهٔ آن و تغییر این پیام پس از گذشتن از مسیری طولانی از فرستاده به گیرنده است.

یکی از روش‌های سادهٔ رفع این مشکل (که بسیار غیربهینه است) استفاده از روشی برای کدگذاری است که به Repetition Code معروف است. در این روش، فرض می‌کنیم پیام c که قصد مخابرهٔ آن را داریم از الفبای 0 و 1 تشکیل شده و حداکثر e بیت هر پیام در هنگام ارسال به دلیل وجود نویز دچار خطا شده و تغییر می‌کنند (بیت 0 به 1 تبدیل شده و برعکس).

می‌توان به سادگی مشاهده کرد که اگر طول c را بزرگ‌تر کرده و هر بیت آن را به جای یک بار، $2e+1$ بار بنویسیم، پیام تولید شدهٔ c' پس از گذشتن از مسیر نویزدار و رسیدن به گیرنده و تبدیل به c''، قابل بازیابی به c است. در بدترین حالت e بیت c' که مربوط به یک بیت c بوده‌اند دچار خطا شده و $e+1$ بیت آن سالم می‌مانند. بنابراین با بررسی اکثریت مقدار بیت‌های c'' که مربوط به هر بیت c هستند، می‌توان به سادگی به هر بیت c دست یافت.

برای مثال اگر حداکثر 1 خطا در پیام ایجاد شود، پیامی مانند `101` را با $2 \times 1 + 1 = 3$ برابر کردن طول پیام، می‌توان از هر نوع خرابی حفاظت کرد. پیام بدست آمده از کدگذاری این رشتهٔ سه بیتی با الگوریتم گفته شده، `111000111` خواهد بود.

حال از شما می‌خواهیم برنامه‌ای بنویسید که این الگوریتم کدگذاری را برای رشته‌های باینری اجرا کند و پیام کدگذاری شده را در خروجی چاپ کند تا آن را با خیالی راحت به گیرندهٔ مورد نظر ارسال کنیم.

ورودی

در اولین خط ورودی یک رشتهٔ باینری به طول حداکثر 100 به شما داده شده که پیامی است که می‌خواهیم آن را کدگذاری کنید.

در خط بعد به شما عدد صحیح e داده می‌شود که ماکسیمم تعداد خطاهایی است که پیام کدگذاری شده در هنگام مخابره با آن روبه‌رو می‌شود.

$$0 \leq e \leq 10^9$$

خروجی

در تنها خط خروجی باید رشتهٔ کدگذاری شده را چاپ کنید.

مثال

ورودی نمونه ۱

```
101011
1
```

خروجی نمونه ۱

```
111000111000111111
```

ماکسیمم تعداد خطاها 1 است. بنابراین لازم است به جای هر بیت، $2 \times 1 + 1 = 3$ بار آن را قرار دهیم. پس به جای هر `0` و `1` به ترتیب `000` و `111` را قرار می‌دهیم.

ورودی نمونه ۲

```
10
4
```

خروجی نمونه ۲

```
111111111000000000
```

ماکسیمم تعداد خطاها 4 است. بنابراین لازم است به جای هر بیت، $2 \times 4 + 1 = 9$ بار آن را قرار دهیم. پس به

جای هر `0` و `1` به ترتیب `000000000` و `111111111` را قرار می‌دهیم.