رمزگذاری دو وجهی

حسام هنوز انسانی توهمی است و فکر می‌کند که نامه‌هایش را به هر پیکی که بدهد آنها نامه را باز می‌کنند و می‌خوانند. لذا این بار روش رمزگذاری دیگری پیدا کرده تا نامه‌هایش را رمز کند. روش جدید حسام به صورت زیر عمل می‌کند.

۱. هیچ چیز به جز کلمات و اعداد را در متن در نظر نگیرید. برای به دست آوردن پس از حذف کردن نقطه، ویرگول، نقطه‌ویرگول و ... جمله باقیمانده را با کاراکترهای سفید (فاصله و اینتر) جدا کنید. دقت کنید که ممکن است بین کلمات بیش از یک کاراکتر سفید باشد.

۲. به هر کلمه یک شماره یکتا نسبت می‌دهد. نحوه شماره‌دهی به این صورت است که از کلمه اول شروع می‌کنیم و زمانی که به یک کلمه می‌رسیم که آن را قبلا ندیده‌ایم، کوچک‌ترین عدد طبیعی را که قبلا نسبت نداده‌ایم را نسبت می‌دهیم. (برای درک بهتر به مثال زیر توجه کنید.)

۳. هر کلمه را با شماره‌اش جایگزین می‌کند. به این شکل حسام می‌تواند آرایه کلمات (که با عدد متناظرشان جایگزین شده‌اند) را به یک پیک و اینکه هر شماره متعلق به چه کلمه ای است را به پیکی دیگر بدهد و از امنیت نامه‌هایش مطمئن باشد.

```python
def encode(text):
    #your code
    return word_to_number_map,numbers
```

▼ راهنمای جداسازی کلمات

برای جدا کردن کلمات با کاراکترهای سفید می‌توانید ابتدا `\n` ها را به ` ` تبدیل کرده سپس از تابع `split` برای جداکردن کلمات استفاده کنید. به این صورت :

```python
text = text.replace('\n', ' ').split()
```

دقت کنید `' '` یک کلمه حساب نمی‌شود و البته توجه کنید که `\n` یک کاراکتر سفید است. به عبارتی همان `' '` است.

▼ تشخیص کاراکترهای اضافی

با استفاده از متد `isalnum` می‌توانید چک کنید یک کاراکتر اضافه هست یا خیر. به طور دقیق‌تر این تابع چک می‌کند که آیا رشته ورودی از حروف (حروف کوچک و بزرگ) و ارقام تشکیل شده یا نه. به مثال زیر دقت کنید:

```
>>> 'Aa12'.isalnum()
True
>>> 'Aa 12'.isalnum()
False
>>> '$@1'.isalnum()
False
```

تابع `encode` را کامل کنید و در فایل `solution.py` قرار دهید و آن را ارسال کنید.

- `numbers` آرایه کلمات است که با شماره‌ها جایگزین شده‌اند (ولی ترتیب باید حفظ شود).
- `word_to_number_map` یک دیکشنری است که کلیدهای آن کلمات و مقدار هر کلید، شماره نسبت داده شده به آن کلمه است.

ورودی نمونه ۱

```
text = """
apple, hello, appl'e, good:  :hello
"""
d , numbers = encode(text)
```

خروجی نمونه ۱

```
({'apple': 1, 'hello': 2, 'good': 3}, [1, 2, 1, 3, 2])
```

ورودی نمونه ۲

```
text = """
:: #ali# $quera$
quera a'l'i ali
quera
"""
d , numbers = encode(text)
```

خروجی نمونه ۲

```
({'ali': 1, 'quera': 2}, [1, 2, 2, 1, 1, 2])
```