یک مینی پیکره فرضی با متن علمی دلخواه، شامل ۵۰ تا ۱۰۰ کلمه انگلیسی(برش از مجله یا web) را در نظر بگیرید. این متن ورودی برنامه است.

برنامه‌ای بنویسید که:

۱- **تعداد unigram و Bigramهای این پیکره را محاسبه و به صورت جدول نشان دهد. (البته با کسر امتیاز جزیی می‌توان این جداول را به صورت دستی به دست آورد.)**

۲- **یک رشته تصادفی از کلمات این پیکره، که طول آن کمتر از ۵ کلمه باشد ایجاد کند و سپس احتمال رخداد این رشته از کلمات را با فرض مینی پیکره کنونی و تقریب Bigram محاسبه کند.**

لینک گیتهاب پروژه : https://github.com/ghazal-pouresfandiyar/nGram-Estimation

کد برنامه حاوی توابع زیر است که طبق اسامی مشخص است هر یک چه کاری انجام می‌دهند اما در ادامه به توضیح برخی از آنها می‌پردازیم. (۶ تابع اول برای قسمت اول سوال و ۵ تابع بعدی برای قسمت دوم سوال می‌باشند.)

```
# calculate ngrams
def ngrams(lst, n):···

# change (tuple as key ---> value) to (srt as key ---> value) for unigram
def change_unigram_format(unigrams):···

def extract_unigrams(file):···

def print_unigrams(unigrams):···

def extract_bigrams(file):···

def print_bigrams(bigrams):···

# probability of hapenning "b a" in sentence
def p(a, b, unigrams, bigrams):···

# count "b a" in corpus
def cba(b, a, bigrams):···

# count b in corpus
def cb(b, unigrams):···

def random_sentence(n, unigrams):···

def calculate_p(test, unigrams, bigrams):···
```

- در هنگام استخراج unigram کلید های دیکشنری به صورت tuple (عضو اول رشته و عضو دوم خالی بود) ذخیره شده بودند که هنگام چاپ جدول ظاهر مناسبی نداشتند به همین دلیل تابع change_unigram_format تعریف شد تا کلید را از حالت tuple به string در بیاورد.
- تابع p احتمال رخداد bigram که به معنای آمدن "$w_{n-1}\ w_n$" به صورت متوالی در جمله است را از رابطه زیر و به کمک توابع cba (برای صورت) و cb (برای مخرج) محاسبه می‌کند:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

- تابع calculate_p احتمال رخداد یک جمله را مشابه مثال زیر پیدا می‌کند:

مثال: احتمال جمله "I want English food" با تخمین bigram

$$P(\text{<s> i want english food </s>})$$
$$= P(\text{i}|\text{<s>})P(\text{want}|\text{i})P(\text{english}|\text{want})$$
$$P(\text{food}|\text{english})P(\text{</s>}|\text{food})$$

البته از آنجا که جملات تصادفی انتخاب می شوند اکثر احتمالات ۰ می‌شوند و احتمالا باید با روشی آنها را smooth کرد.

نمونه‌ی اجرا شده :

پیکره :

a.txt

```
<s> I am Sam </s>
<s> Sam i am </s>
<s> I do not like green eggs and ham.</s>
```

خروجی:

```
-----------------Unigram values-----------------
              <s> ---> 3
              i ---> 3
              am ---> 2
              sam ---> 2
              </s> ---> 3
              do ---> 1
              not ---> 1
              like ---> 1
              green ---> 1
              eggs ---> 1
              and ---> 1
              ham ---> 1
-----------------Bigram  values-----------------
          ('<s>', 'i')  ---> 2
          ('i', 'am')  ---> 2
          ('am', 'sam')  ---> 1
          ('sam', '</s>')  ---> 1
          ('<s>', 'sam')  ---> 1
          ('sam', 'i')  ---> 1
          ('am', '</s>')  ---> 1
          ('i', 'do')  ---> 1
          ('do', 'not')  ---> 1
          ('not', 'like')  ---> 1
          ('like', 'green')  ---> 1
          ('green', 'eggs')  ---> 1
          ('eggs', 'and')  ---> 1
          ('and', 'ham')  ---> 1
          ('ham', '</s>')  ---> 1
The test sentences is : <s> and sam am not am </s>
p(and|<s>) =  0.0
p(sam|and) =  0.0
p(am|sam) =  0.5
p(not|am) =  0.0
p(am|not) =  0.0
p(</s>|am) =  0.0
total = 0.0
```

پیکره:

corpus.txt

```
<s> I am Sam </s>
<s> Sam i am </s>
<s> I do not like green eggs and ham</s>
<s> the quick person did not realize his speed and the quick person bumped </s>
<s> i am ghazal prs and ghazal prs is me</s>
```

خروجی:

```
------------------Unigram values------
                <s> ---> 5
                i ---> 4
                am ---> 3
                sam ---> 2
                </s> ---> 5
                do ---> 1
                not ---> 2
                like ---> 1
                green ---> 1
                eggs ---> 1
                and ---> 3
                ham ---> 1
                the ---> 2
                quick ---> 2
                person ---> 2
                did ---> 1
                realize ---> 1
                his ---> 1
                speed ---> 1
                bumped ---> 1
                ghazal ---> 2
                prs ---> 2
                is ---> 1
                me ---> 1
```

```
------------------Bigram  values------------------
          ('<s>', 'i')  ---> 3
          ('i', 'am')  ---> 3
          ('am', 'sam')  ---> 1
          ('sam', '</s>')  ---> 1
          ('<s>', 'sam')  ---> 1
          ('sam', 'i')  ---> 1
          ('am', '</s>')  ---> 1
          ('i', 'do')  ---> 1
          ('do', 'not')  ---> 1
          ('not', 'like')  ---> 1
          ('like', 'green')  ---> 1
          ('green', 'eggs')  ---> 1
          ('eggs', 'and')  ---> 1
          ('and', 'ham')  ---> 1
          ('ham', '</s>')  ---> 1
          ('<s>', 'the')  ---> 1
          ('the', 'quick')  ---> 2
          ('quick', 'person')  ---> 2
          ('person', 'did')  ---> 1
          ('did', 'not')  ---> 1
          ('not', 'realize')  ---> 1
          ('realize', 'his')  ---> 1
          ('his', 'speed')  ---> 1
          ('speed', 'and')  ---> 1
          ('and', 'the')  ---> 1
          ('person', 'bumped')  ---> 1
          ('bumped', '</s>')  ---> 1
          ('am', 'ghazal')  ---> 1
          ('ghazal', 'prs')  ---> 2
          ('prs', 'and')  ---> 1
          ('and', 'ghazal')  ---> 1
          ('prs', 'is')  ---> 1
          ('is', 'me')  ---> 1
          ('me', '</s>')  ---> 1
The test sentences is : <s> prs ghazal is me do </s>
p(prs|<s>) =  0.0
p(ghazal|prs) =  1.0
p(is|ghazal) =  0.0
p(me|is) =  0.0
p(do|me) =  0.0
p(</s>|do) =  0.0
total = 0.0
```