Python Prerequisite Python Programming Test

0. Programming Environment:

Python 3.6 (Recommended), Python 2.7

1. Reminder

Write a program to calculate the result of $\mathbf{a}^{\mathbf{k}!} \% \boldsymbol{p}$. For example

When a = 2, k = 2, p = 3, the answer will be 2 ^ (2!) % 3 == (2 ^ 2) % 3 => 1
When a = 3, k = 3, p = 5, the answer will be 3 ^ (3!) % 5 == (3 ^ 6) % 5 == 729 % 5 => 4

Therefore, what's the answer of following two cases:

Case1: a = 112, k = 13, p = 79, ans = ?

Case2: a = 12345678, k = 5678910, p = 1234567890, ans = ?

2. Sentence Candidates Parser

In a QA system, an ordinary task is to give some Question-Answer Pair to dataset. For example, we may give dataset a pair (“你好”, “你好，我是网银小助手”), or (“如何办理网银盾”， “您在网页下方点击 XXX 然后进行下一步即可”).

Actually, this QA pair could give sufficient information to a QA system. However, there are some occasions that need some more sophisticate processing. For example, someone may give (“如何办理贵宾卡/金卡/特惠卡”, “请在页面下方 XXX 处点击办卡”)， or (“网银/信用卡如何/怎样注销/开户”, “请在账户管理处进行”). Therefore, we need a parser, given a sentence with ‘split mark’, this parser will generate some more sentences:

e.g:

Input: 如何办理贵宾卡/金卡/特惠卡?
Output:
    如何办理贵宾卡?

    如何办理金卡?
    如何办理特惠卡?

Input 2: 网银/信用卡如何/怎样注销/开户?
Output:
    网银如何注销?
    信用卡如何注销?
    网银怎样注销?
    信用卡怎样注销?
    …
    信用卡怎样开户?
    # above should exist 8 sentences.

Please write a program to solve this problem. Hint: You may need the python package “**jieba**” https://github.com/fxsjy/jieba to split the sentence to corresponding words.

3. Random Chinese Sentence Generator

Writing a programming which could generate random Chinese sentences based on one grammar.

Your input grammar is:

simple_grammar = """
sentence => noun phrase verb_phrase
noun_phrase => Article Adj* noun
Adj* => null | Adj Adj*
verb_phrase => verb noun_phrase
Article => 一个 | 这个
noun => 女人 | 篮球 | 桌子 | 小猫
verb => 看着 | 坐在 | 听着 | 看见
Adj => 蓝色的 | 好看的 | 小小的 """

Your task is to define a function called *generate,* if we call generate('sentence'), you could see some sentences like:

>> generate("sentence")

Output: 这个蓝色的女人看着一个小猫

>> generate("sentence")

Output: 这个好看的小猫坐在一个女人

***Bonus*: Can you modify your programming code, such that, if we change the grammar, we don't need to change source code at all, this program will generate sentence appropriately?**

Of course, you may create more complicated grammar if you like.