Python Prerequisite Python Programming Test

祝贺你，你已经马上要成为一名真正的算法工程师，不，准确得说，应该是一名人工智能工程师了。而且我们将要进行的是最有趣，最具有挑战性，也是使用最广的自然语言理解方向。希望大家已经做好了准备。

但是，第一步，为了使得我们之后的课程能够合理得继续，需要对大家的编程水平进行测试。请大家在以下三道题选择一道，完成编码并在2018年11月1日前提交。

但是如果这三道题目对你而言都太过困难，你可以先自学预修课程：，https://cn.udacity.com/course/intro-to-computer-science--cs101，这个课程设计的非常好，大约需要大家20 – 30个小时的学习时间。

可能你已经注意到了，我们的这三道题目，一道是关于数值计算/模拟计算的，两道是关于语言解析的。这三道题目每一道都是很有趣的，也是我们以后工作中会经常遇到的问题的一个缩影，希望大家能花时间解决掉。

有同学说，如果预科课程对自己来说也很难，也有问题，该怎么办？为了保证我们各个顾问的时间，大家可以先交费，我们开始预科学习，我们会组织一个微信群，由我们的顾问为大家解答。在正式课程（11月3日）开始的时候，如果大家没有完成预科课程，我们会在11月3日当天，退还全部学费给您。

好的，开始我们的激动的学习之旅吧!

0. Programming Environment:

Python 3.6 (Recommended), Python 2.7

1. Spiral Memory

You come across an experimental new kind of memory stored on an infinite two-dimensional grid.

Each square on the grid is allocated in a spiral pattern starting at a location marked 1 and then counting up while spiraling outward. For example, the first few squares are allocated like this:

```
17  16  15  14  13
18   5   4   3  12
19   6   1   2  11
20   7   8   9  10
21  22  23---> …
```

While this is very space-efficient (no squares are skipped), requested data must be carried back to square 1 (the location of the only access port for this memory system) by programs that can only move up, down, left, or right. They always take the shortest path: the Manhattan Distance between the location of the data and square 1.

For example:

Data from square 1 is carried 0 steps, since it's at the access port.
Data from square 12 is carried 3 steps, such as: down, left, left.
Data from square 23 is carried only 2 steps: up twice.
Data from square 1024 must be carried 31 steps.
How many steps are required to carry the data from the square identified in your puzzle input all the way to the access port?

Your Puzzle Input is: 2345678
Your Answer is:

2. Sentence Candidates Parser

In a QA system, an ordinary task is to give some Question-Answer Pairs to a dataset. For example, we may give 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”) to a dataset.

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’, which 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 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, this program will generate sentence appropriately.

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

好了，看完这些题目，如果你已经有了想法，知道怎么做，那真是太好了，你已经具备了所有的基础条件，就差一些具体的 AI/机器学习算法知识了。那么，该如何提交呢?

提交文件包括:

1. 选择问题的样例输出;
2. 该问题的源代码;

提交答案：请于10.31之前，按照【submission】+ 姓名 的格式回复至 deeplearning.nlp.zh@gmail.com 该邮箱，源代码请用附件的形式提交;

答疑环境：若有问题，可以直接回复邮件（deeplearning.nlp.zh@gmail.com），或可以关注公众号："学算法的小黑狗"，直接回复问题。

如果你不知道怎么做，那也没关系，请先上这门非常优秀的免费课程，https://cn.udacity.com/course/intro-to-computer-science--cs101，上完之后，就可以参加我们接下来的课程了。

好的，那我们开始吧~ 期待我们的再次相遇。