

入学 Python 编程能力测试

祝贺你，你已经马上就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或者计算机视觉方向。文字和图像是我们日常输入最多的两种信息，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，完成编码并且测试输入和输出，观察程序的运行是否符合程序的预期。

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，两道是关于数值计算/模拟计算的，一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一个缩影，希望大家能花时间解决掉。不论你是选择的自然语言处理还是计算机视觉，这些题目你都可以做。这些程序考察的是编程能力，和具体的背景知识无关。

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（2019年3月31日）开始的时候，没有完成预科课程的同学，我们会在3月31日当天，退还全部学费给您。您可以继续学习预修课，在下一期课程参加学习。

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?

How to test your answer:

If you input: **100000** Your Answer should be: **173**

If you input: **2345678** Your Answer should be: **1347**

2. Simple Number Finding

You are playing a card game with your friends. This game in China named “扎金花”. In this game, the 2, 3, 5 are some simple powerful numbers. Because the combination of 2,3,5 is less than any other combinations but greater than the AAA, which is the king in this game. In today, you want to find if a number is a simple number, in which their factors only include 2, 3 and 5.

So your task is to find out whether a given number is an amazing number.

E.g

Input: 6

Output: (2, 3)

Explanation: $6 = 2 \times 3$

Input: 8

Output: (2, 2, 2)

Explanation: $8 = 2 \times 2 \times 2$

Input: 14

Output: None

Explanation: 14 is not amazing since it includes another prime factor 7.

How to check your answer:

If you test **1845281250**, your program should give (2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5)

If you test **3690562500**, your program should give (2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5);

If you test **1230187500**, your program should give (2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5);

If you test **10023750**, your program should give **None**;

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 | 听着 | 看见
Adj => 蓝色的 | 好看的 | 小小的 | 年轻的
""
```

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

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，你已经具备了基础条件，就差一些具体的 AI/机器学习算法知识了。

如果你的程序运行结果与我们的给出的例子一致（第三题是随机函数，应该产生类似的句子），那么请检查课程相关协议是否有问题，然后将协议签名（推荐使用“好签”app 进行电子签名）之后返回至邮箱 deeplearning.nlp.zh@gmail.com。

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，然后完成这三道题中的任意一道，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。

