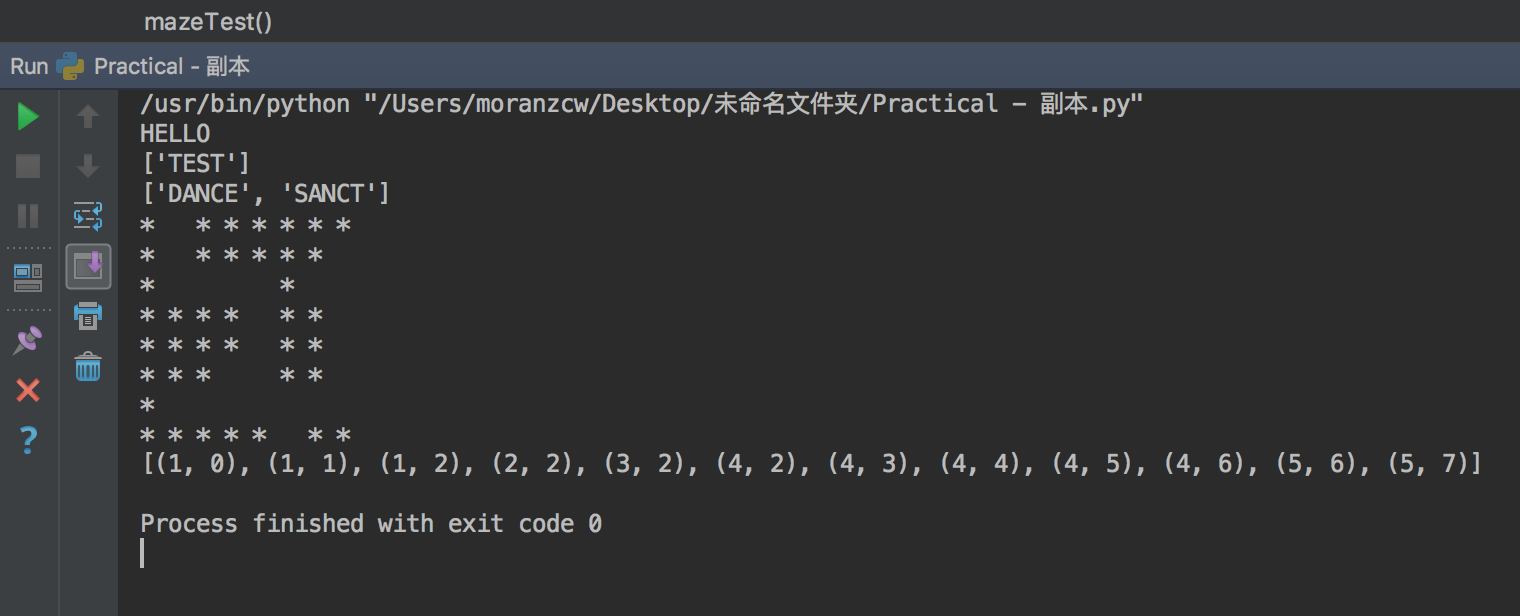
1. 在Python 2下直接运行。
2. 实现了要求的所有函数，mazeTest() 测试函数最后添加了两行，用于测试Maze类中的printMaze函数和route函数
3. 运行后输出如下：



1. 调用morseDecode函数，参数为HELLO的morse码，['....','.','.-..','.-..','---']，输出“HELLO”
2. 调用morsePartialDecode函数，参数['x','x','x..','x']和['x..','x-','x.','x.-.','x']， 输出[“TEST”]和[“DANCE”,”SANCT”]
3. 调用printMaze函数，显示迷宫形状
4. 调用route函数，参数为1,0,5,7, 输出(1,0)至(5,7)的路径
5. 实现细节

a) Morse Code

这一问的目的是把摩斯码代表的单词翻译出来，输入为一个字符串列表，每个字符串代表一个字母，所有字符串合起来就是一个单词。

用Python的Dict类型存储摩斯码到字母的映射，key为摩斯码，value为字母。

如果单次查找映射表的时间复杂度为O(1), 输入的字符串列表长度为n(即包含n个字符串)，那么此函数的时间复杂度为O(n),空间复杂度为O(1)。

b) Incomplete Morse Code

这一问目的同上，但代表每个字母的摩斯码字符串中，第一位未知，查找所有合法的字母组合以构成合法的单词。

此算法分为两步：

1. 枚举所有可能的组合（无论是否合法），每个未知字符可能是'.'或'-'，对于长度为n的字符串列表，共n个未知字符。枚举所有组合的时间复杂度为O(2^n)，空间复杂度为O(2^n)。

2. 在合法词汇表中查找每个已枚举的组合，将合法的组合作为结果返回，时间复杂度O(1)，空间复杂度O(1)。

总时间复杂度为O(2^n)，空间复杂度为O(2^n)。

c) The maze

1. 函数addCoordinate(x,y,blockType)

向迷宫中添加墙面或者空地，当坐标(x,y)超出迷宫尺寸时，函数会扩展迷宫尺寸以包含坐标(x,y)，扩展的部分默认为墙面。

2. 函数printMaze()

打印迷宫形状。

3. findRoute(x1,y1,x2,y2)

计算坐标(x1,y1)到(x2,y2)的路径，返回一条包含路径上所有坐标的list，若路径不存在，返回空list。

算法采用回溯法实现，findRoute调用route函数，route函数递归调用自己。从一个点出发，按上，下，左，右的相邻点依次尝试，若发现该该点没有路径可尝试，则退回到上一步，直到到达目标或搜索完所有路径。

若迷宫尺寸为m\*n，因为使用变量triedSet来避开已尝试的坐标，所以时间复杂度为O(m\*n)，空间复杂度为O(m\*n)。