**任务要求：找成语游戏**

在7\*7的方格，随机产生n个成语，每个成语横或者竖的连续放到格子中，非成语部分随机填汉字，设计游戏为三个等级，等级越高填充成语数目越多。程序执行时，选择一个等级，生成一套填充，结果以html文件形式可在浏览器中显示。

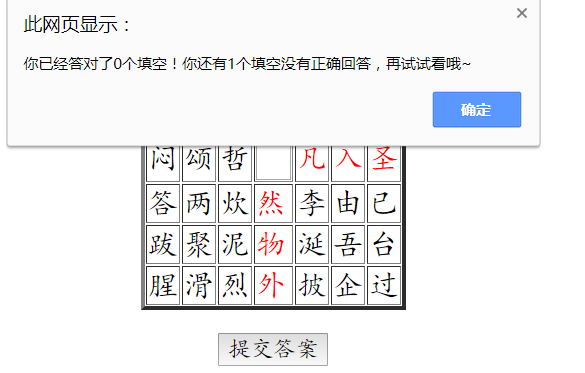
**3个等级的网页分别为level1.html、level2.html、level3.html，成语显示为红色**

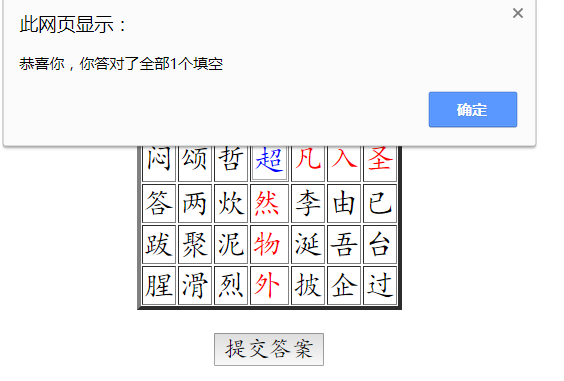
**对应的文本文件为：answer1.txt、answer2.txt、answer3.txt**

**部分运行结果如下：**

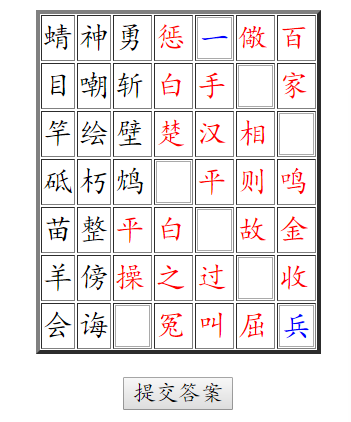
**等级1：**

****

****

****

**等级2：**

****

****

**等级3：**

****

****

1. **设计思路及算法说明**

**一．读取文件**

1.利用正则表达式模式匹配取出文本文件中的一个成语。

2.利用/../g模式匹配这个成语中的3或4个汉字存入一个空数组中。（文本文件中的“杀风景”一词为3个字）

3.遍历刚赋值完毕的有3或4个元素的数组，以该成语为关键字，构建一个值为数组的哈希$HashHZ{$Idiom}，并将该成语中含有的汉字依次存入数组@{$HashHZ{$Idiom}}中。

4.遍历的同时将以每个汉字为关键字的哈希值置1，保证之后建立汉字数组时数组中的元素没有重复。

5.遍历时取正在处理的字在成语中的下标，将这个字所在的成语及字在成语中的下标一起存入数组@HzInIdiom，即这个数组仅有2个元素，第一个为汉字，第二个为汉字在成语中的下标。

6.以汉字为关键字构建值为数组的哈希$HashIdiom{$HZ}，以硬引用的方式将@HzInIdiom存入@{$HashIdiom{$HZ}}，即先构建一个值为数组的哈希，再向这个数组中添加元素，添加的元素也是数组。内含数组即为在第5条中构建的数组。

7.以上几条在读文件时完成。遍历第4条构建的哈希的关键字存入数组@ArrHZ，这个数组中的元素即为所有成语中出现的字（不重复）。

**二．获取等级**

在命令窗的简单交互，选择1-3中的某个数字作为等级。当输入非法时则要求用户重新输入。

**三．生成填充**

**说明：这是程序的核心部分。主要是先随机生成成语，而后成语的位置是在程序中人为设定的。在此前曾经尝试过成语以及成语的位置都随机，这种情况随机的变量太多，鲁棒性比较低，在输出多个成语时程序会崩溃，因此改变策略，采用自己卡位置，并且为了防止生成的不同等级的网页显得缺乏变化，成语依次输出的位置也有意进行了调整。**

1. 等级1（理论上最多有4个成语，具体视实际情况而定）：



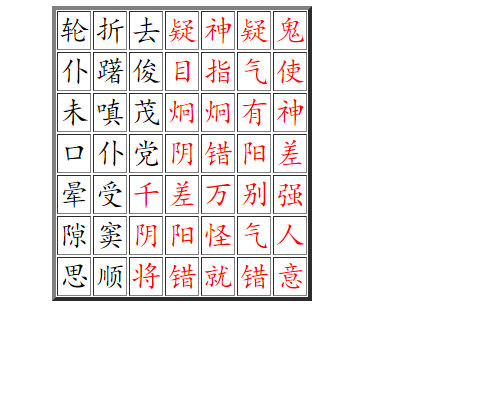
1. 定义空数组@IdiomArray用于存放依次输出的成语并记录每个成语的下标，当找不到符合条件的成语时，则将“ ”压入数组@IdiomArray，保证后来再生成成语时选择的参照成语下标正确。
2. 定义空数组@Array1为二维数组的数组，实际作用为计算总计输出的成语数，因为最终@Array1中的几个变量相同。
3. 生成第一个成语：在汉字数组中随机取一个汉字，随机取一个在上一步第六条中生成的数组@{$HashIdiom{$HZ}中的元素，该元素为一个数组，该数组中的元素为取到的汉字所在的某个成语和这个字在该成语中的下标组成的数组。将这个数组中的成语元素存入@IdiomArray中并返回这个成语中的几个字组成的汉字数组。将第一个成语的第一个字所在的行和列定为第4行第4列，分别以行和列的下标为二维数组的元素，依次给二维数组的元素赋值，之后在网页中横向输出第一个成语。
4. 生成第2到第4个成语：

2：以第1个成语为参照取一个成语，该成语的首字与第1个成语的首字相同，则该成语首字的位置也为第4行第4列，依次给二维数组的元素赋值，竖向输出。

3：以第1个成语为参照取一个成语，该成语的首字与第1个成语的尾字相同，则该成语首字的位置为第4行第7列，依次给二维数组的元素赋值，竖向输出。

4：以第1个成语为参照取一个成语，该成语的尾字与第1个成语的尾字相同，则该成语首字的位置为第1行第7列，依次给二维数组的元素赋值，竖向输出。

2． 等级2（理论上最多有10个成语，具体视实际情况而定）：



a. 运行一次等级1，将得到的数组赋值给@Array1, 当等级1生成的成语数小于4时，则重新运行等级1，避免等级间没有区分度的情况。

b. 生成第5到第10个成语：

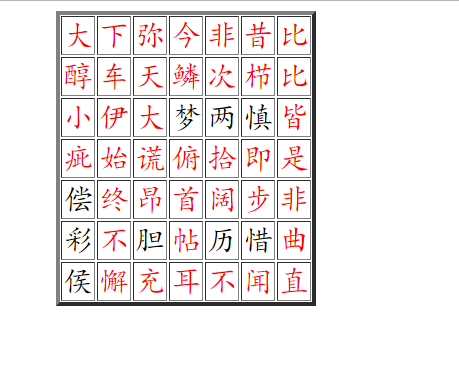
5-6：以第2个成语为参照取一个成语，这两个成语的第2个字分别与第2个成语的第二、三两个字相同，第5个成语的首字位置为第5行第3列，第6个成语的首字位置为第6行第3列，依次给二维数组的元素赋值，横向输出。

7：以第4个成语为参照取一个成语，该成语的尾字与第4个成语的首字相同，则该成语首字的位置也为第1行第4列，依次给二维数组的元素赋值，横向输出。

8：以第2个成语为参照取一个成语，该成语的首字与第2个成语的尾字相同，则该成语首字的位置也为第7行第3列，依次给二维数组的元素赋值，横向输出。

9-10：以第4个成语为参照取一个成语，这两个成语的第4个字分别与第4个成语的第二、三两个字相同，第5个成语的首字位置为第2行第4列，第6个成语的首字位置为第3行第4列，依次给二维数组的元素赋值，横向输出。

1. 等级3（等级3理论上最多可以生成15个成语，具体要视实际情况而定）：



a.运行一次二级。

b.生成第11-15个成语：

11-14：第11和第13个成语的生成大部分与等级1中生成第一个成语的步骤相同，不过最后为纵向输出。第11个成语的首字位置为第1行第一列，第13个成语的首字位置为第1行第二列。

第12个成语和14个成语分别以第11个成语和第13个成语为参照取一个成语，这两个成语的第1个字分别与第11个成语和第13个成语的最后一个字相同，第12个成语首字在第4行第1列，第14个成语首字在第4行第二列，依次给二维数组的元素赋值，纵向输出。

第15个成语与第11个成语生成基本相同，纵向输出，首字位置为第一行第三列。

**四．输出结果**

1.输出网页：根据选定的等级，执行对应的子函数。遍历返回的二维数组，如果是汉字则标记位置。再次遍历，每输出四个汉字设置一个空格供用户填充，否则从汉字数组中随机取汉字输出。

2.输出文本：遍历@IdiomArray，如果是成语则输出。成语的总个数为@Array1的数值。

**b. 程序注释，主要代码给出注释（具体可见源程序）**

由于只有printout函数与之前相比有变动，这里只贴这部分代码，其他部分的可以查看源程序或者是之前静态版的总结报告。

****

****