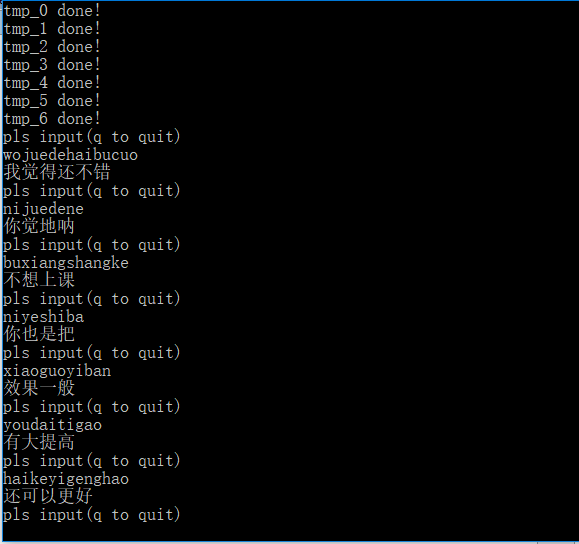
总结

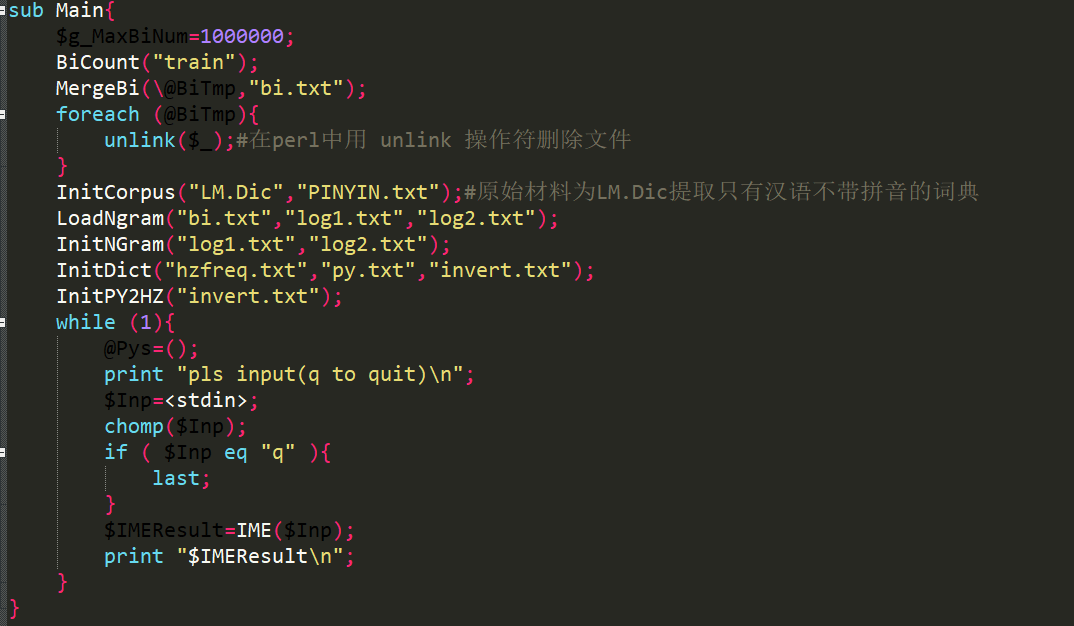
任务需求：给定语料训练Bigram模型，利用该模型，完成基于句的拼音输入法，输入拼音为不带空格、无调的全拼序列。

用到的原始语料： LM.Dic、py.txt、hzfreq.txt、train。

运行效果：

：

思路：



BiCount与MergeBi函数采用分治处理方法统计train语料中两个汉字两个汉字接连出现的次数输出到bi.txt中，为下一步计算常用汉字的一元和二元概率对数值做准备。

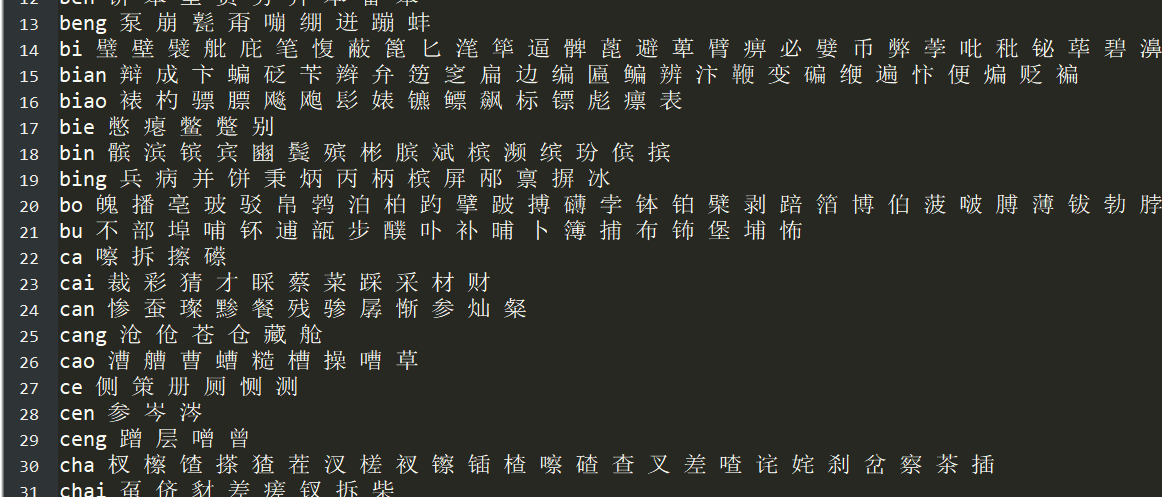
InitCorpus函数将LM.Dic词典中的拼音提取出来，为后面输入不带空格、无调的全拼序列时最大长度匹配做准备。

LoadNgram利用bi.txt中的二元数据计算常用汉字的一元和二元概率对数值，分别输出到log1.txt与log2.txt中。

InitNGram读入log1.txt与log2.txt中的对数值数据，将一元概率读入%HashUni，将二元概率读入%hashBi。

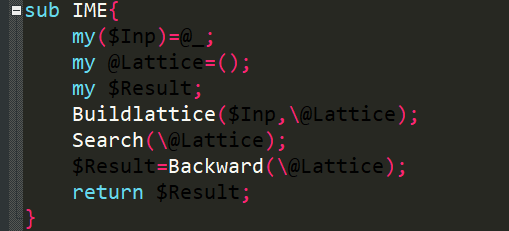
InitDict利用老师给出的字频文件hzfreq.txt及拼音文件py.txt得到拼音-》汉字文件。其中每个拼音对应的汉字按字频由大到小输出到invert.txt中。

invert.txt如下：



InitPY2HZ初始化拼音与候选汉字对照表，为后续得到候选汉字做准备。

在Main中接收用户输入的拼音，调用IME()，转为汉字。

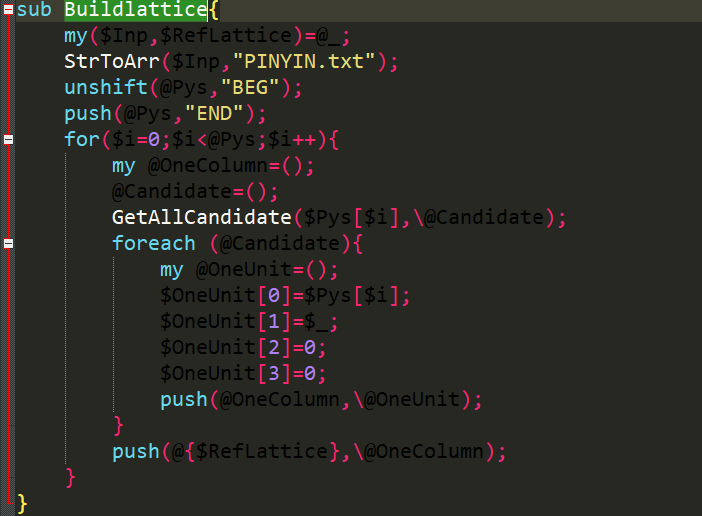


IME函数为音字转换过程的主函数，函数输入为不带空格、无调的全拼拼音，函数返回转换结果。主要包括三个步骤：

调用Buildlattice，建立候选网格。

调用Search，从左到右为网格单元赋值。为单元计算累计最大概率值以及指向获取最大值的前一个单元的回退指针。

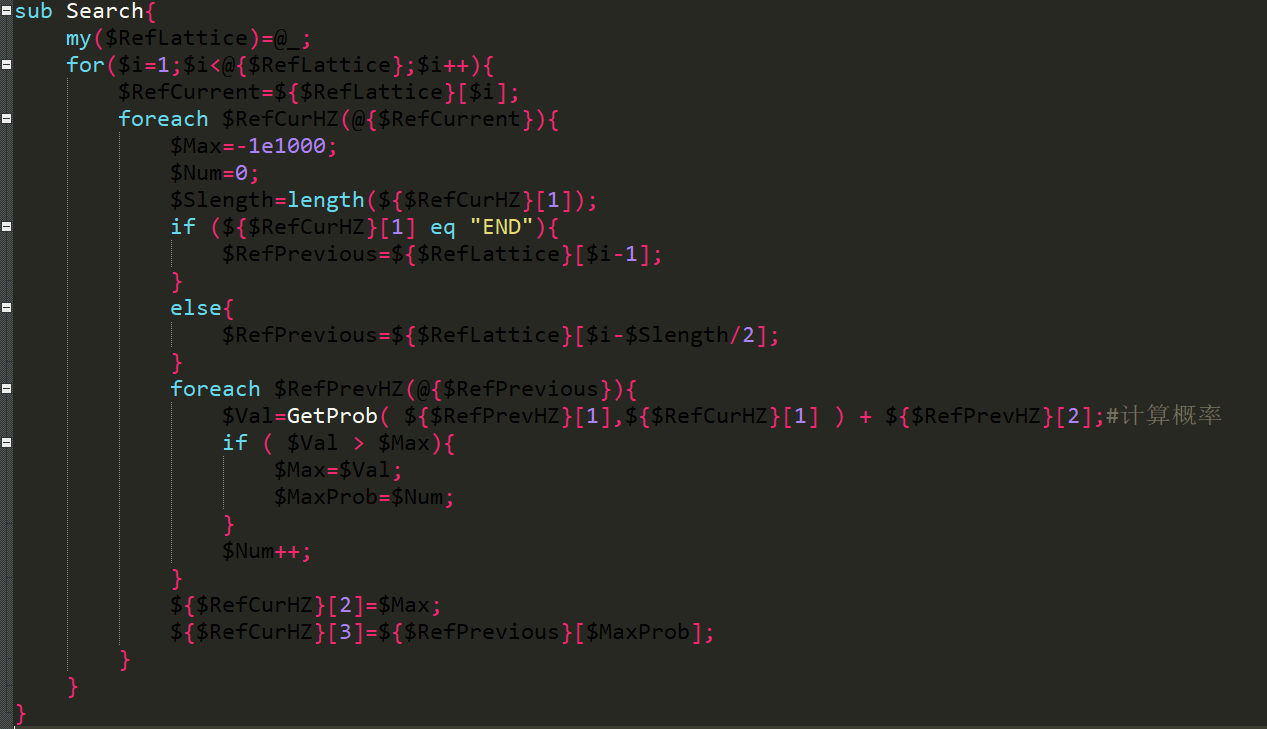
调用Backward，从网格最右单元开始，依据回退指针获得转换结果。



建立候选网格，$Inp在StrToArr中利用最大长度匹配算法转换为数组@Pys。

对@Pys的每个成员建立一个子数组，其成员也是一个数组，即候选网格单元。其中第一个成员为拼音，第二个成员为候选汉字，第三个成员为概率累计值，第四个成员为回退指针。

调用GetAllCandidate()，取得一个拼音所有可能转换的候选汉字。



该函数为每个单元累计概率并获取回退指针。为调用Backward()进行回退做准备。函数输入参数为指向候选网格数组的引用。

该函数通过回退取得最后的音字转换结果：

