**项目四 中文全拼输入法**

**一、实验题目分析**

尝试完成一个带记忆功能的全拼输入法，即：从键盘输入一个全拼的拼音序列，输出拼音的切分结果，根据项目提供的数据文件给出拼音序列对应的字或词的候选。

1.完成拼音的切分（默认按照正向最大匹配）

2.切分后显示候选词

3.用户输入的词存入用户词典

**二、需求分析与设计思路**

1.数据的读取与存储

读入

我们注意到piyin.txt中每一行都是由拼音+“，”+一串候选字组成的，而候选字显然不是按照词频进行排序的，因此我们要先读入项目三中的1字串文件，把它存储在结构变量数组oneword中（oneword中含有string 和词频num），

我们不妨构造一个结构变量pinyin，由string p 表示拼音 oneword型数组 character[110]存储候选字。然后我们利用项目三中的方法（见到字母储存到p，见到“，”跳过，见到汉字储存到character[i]），之后根据词频对读入的pinyin的数据进行排序。

把排序的代码封装在Createandsort函数中。

2.拼音的切分

与项目三类似，在主函数中读入拼音串，存入到char类型的line数组中，然后再对line进行切分。定义一个Pin函数做这项工作。

我们注意到，一个中文拼音字母可以大致分为三类，声母（只能用来开头），韵母（可以结尾，也可以放在中间，有时还可以放在开头），以及n, g ,r(可以开头或结尾)

那么对于line中的拼音，我们先读取一个到pin[j]中

它可以分为三类：声母开头、韵母开头和分隔符“\”，还有一类特殊的”tx”这是T恤衫的拼音，我们把它单独拿出来处理

1）当line[i]是可以单独作为一个拼音的开头的韵母（a, o, e）时，我们接着判断下一个字符是否与它一起构成an ao ai ei en er o ou on，如果不是，i++,j++进行下一次循环，如果是，进行下一个判断，判断第三个是否与它一起构成 ang eng ong 如果不是 把第二个存到pin里，如果是，把第二第三个都存到pin里

然而，经过测试，我发现一个问题，之前方法默认是按照正向最大匹配，然而对于aou anv anu 这样的拼音，虽然不存在歧义，但是按照正向最大匹配会被分成an’i 这显然不对，因为u I v 这样的字母并不是有效的拼音，所以我们要在每种情况中再判断一下它是否要按照正向最大匹配。

2）当line[i]是声母（包括ngr）时，处理方法与韵母类似，只不过line[i]要往pin中多存一个字符，并且拼音字母的组合要比韵母开头的拼音要多，若用\*代替声母

两个字符的拼音：\*a, \*o, \*e, \*i, \*u, \*v

三字符：an ai ao ei en in ia ie io iu ua uo ue ui un ou on

四字符：ang eng ong ing iao ian uai uan ion

五字符：iang iong

在这里，我们依然要考虑不是正向最大匹配的情况，讨论如下：

三个字符的拼音，用\*代替声母

ia ie iu in

ia: ia\* 不变

iai 要处理

ie: ie\*不变 ier\*要处理 ien\*要处理

iei要处理

iu不变

in:

in\* ina不变 ini inu inv 要处理 ino ine讨论

inong ineng要处理

ua ue ui un uo

ua

ua\* uaa uai 不变

uao 只有uaou时不需要处理

ue

ue\*不变 ueg uea uee不变 uen uer ueo 讨论 uei处理

uen\*处理 其余不处理

uer\*处理 uera 处理 其余不处理

ui和uo不变

un

同in

四个字符的拼音：

ing eng 同ang

ong 不变

iao 只有u时处理

ian\*(包括ngr) iana 不变 iane iano 讨论 ianu ianv iani 处理

ianeng ianong 时处理

uai不变

uan同ian

声母四个

iang iong uang

iong不变

iang uang 同韵母

ang ing ong

ong 不变

ang:

ang\* anga不变 angu 一定要处理

ange ango 讨论

ange:只有angeng要处理

ango 只有angon(g)要处理

注：图中如果标注“要处理”，那么相较于之前要少存一个line中的字符

同时，我们应当考虑到zh ch sh 这种声母，这时应该把line[i]存到pin中，line移到下一个再进行判断。

3) 分割符“\”

不进行储存line 和pin都移到下一个

4）line中出现tx

line移到下一组拼音字符的开头，pin中储存txushan 单独处理

做完这些以后在读入的pinyin数组中寻找pin对应的拼音，，建立新的结构变量数组intpin，这样我们对于输入的字符串就找到了它的拼音切分和对应的候选字列表。

3.通过拼音切分找词——用两个单独的函数builtword读入数据，dealwithword进行候选词的匹配

首先我们应当把word.txt中的文件读入，注意，我们要从第二个开始读入，因为第一个词是“T恤衫”，我们单独来处理

然后对于word中的每个词，假如能在intpin的候选字列表中找到与它相同的字，那么这个词就是候选词，然后按照字的长短进行排列，存到word类型数组choice里

4.输出到屏幕以及建立用户词典

把输出功能封装在专用的show函数中。

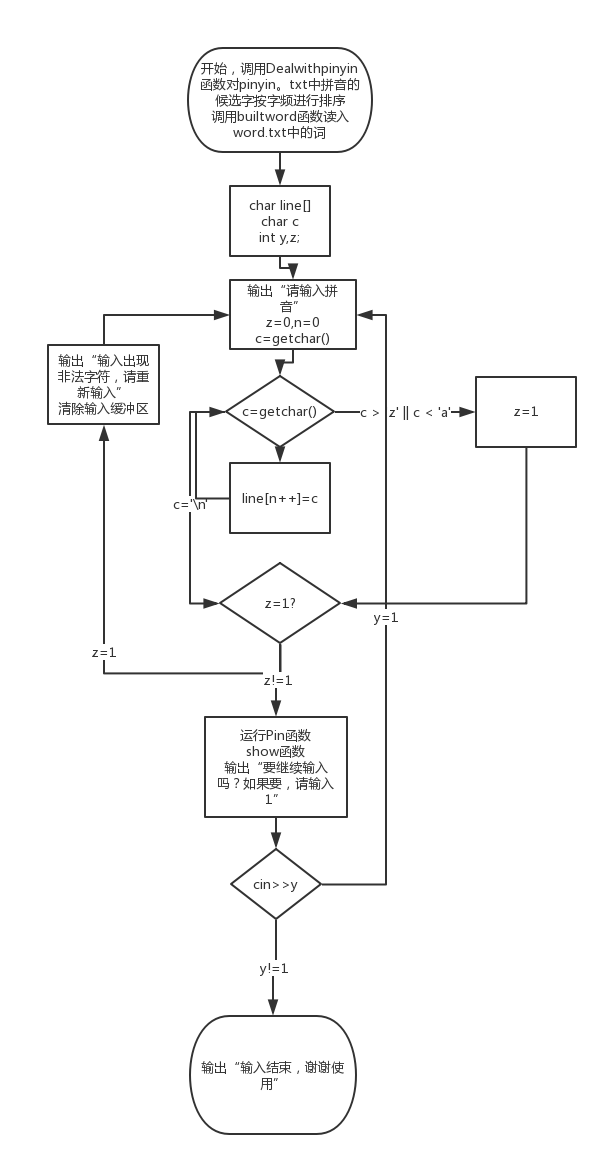
先读入用户词典，处理方式与word.txt相同，存入choice数组

如果没有用户词典先建立用户词典，输出一个“用户词典”（目的是为了在处理时与word.txt统一，因为第一个词要跳过）。

先输出choice数组中的词，即候选词，然后输出当前拼音对应的候选字，把用户的选择存入一个string类型变量中，输出“最终结果：s”，再把s输出到用户词典中。

**三、主函数流程图**

主函数主要负责读入原始的拼音串并把它存入line数组中，以及各种数据的读入。



**四、各功能函数说明**

**1.对pinyin.txt文件进行处理的函数dealwithpinyin**

**先读入项目三中的1字串文件，把它存储在结构变量数组oneword中（oneword中含有string 和词频num），**

**构造一个结构变量pinyin，由string p 表示拼音, oneword型数组 character[110]存储候选字。然后我们利用项目三中的方法（见到字母储存到p，见到“，”跳过，见到汉字储存到character[i]）。**

**之后调用函数createandsort根据词频对读入的pinyin的数据进行排序，**

代码如下：

void dealwithpinyin(pinyin Pinyin[])

{

m = 0;

ifstream in("1字串.txt");

if (!in)

{

cout << "打开文件失败" << endl;

}

oneword t;

while (in >> t.s) {

in >> t.num;

st[m++] = t;

}

char str[2000];

char c;

int n, k, s;

ifstream infile("pinyin.txt");

if (!infile)

{

cout << "打开文件失败" << endl;

}

k = 0;

for (; ; ) {

s = 0;

n = 0;

for (; ; ) {

if (infile.eof()) { s = 1; break; }

else {

infile.get(c);

if (c == '\n') break;

else if (c >= 0 && c <= 127) { if (c>='a'&&c<='z')Pinyin[k].p += c; continue; }

else str[n++] = c;

}

}

str[n] = '\0';

Createandsort(str, Pinyin[k], n);

k++;

if (s == 1) break;

}

}

Createandsort函数：

void Createandsort(char str[], pinyin &pin, int n)

{

int i, j, k;

for (i = 0, j = 0; i < n; j++) {

pin.character[j].s = str[i++];

pin.character[j].s += str[i++];

pin.character[j].num = 0;

for (k = 0; k < m; k++) {

if (st[k].s == pin.character[j].s) { pin.character[j].num = st[k].num; break; }

}

}

pin.character[j].s = "0";

sort(pin.character, pin.character + j, cmp);

}

**2.读入word.txt中数据的函数builtword,按行读，把每一行存到word 类型的结构变量数组中，在这里每一个结构变量包含数个oneword，每个oneword表示词中的一个中文字。**

void builtword(pinyin Pinyin[], word total[],int &n1)

{

ifstream in("word.txt");

if (!in)

{

cout << "打开文件失败" << endl;

}

word t;

char s[40];

int i, j, k, l;

n1 = 0;

string temp;

in >> s;//跳过T恤衫

while (in >> s) {

j = 0; l = 0;

for (i = 0; s[i]; i = i + 2) {

temp = s[i]; temp += s[i + 1];

t.str[j] = temp; j++;

}

t.sum = j;

total[n1++] = t;

}

}

**3.对拼音串进行切分的函数pin，具体思路已经在需求分析中给出。**

void Pin(char line[], int &n, pinyin intpin[])

{

int i, j, s;

string pin[30];

for (i = 0, j = 0, s = 0; line[i] != '\0'; ) {

if (s == 0)pin[j] = line[i];

s = 0;

if (line[i] == 'a' || line[i] == 'e' || line[i] == 'o') {//假如第一个字符是韵母

if (!((line[i] == 'a' &&line[i + 1] == 'i') || (line[i] == 'a' &&line[i + 1] == 'o') || (line[i] == 'a' &&line[i + 1] == 'n')

|| (line[i] == 'e'&&line[i + 1] == 'i') || (line[i] == 'e'&&line[i + 1] == 'n') || (line[i] == 'e'&&line[i + 1] == 'r')

|| (line[i] == 'o' &&line[i + 1] == 'u') || (line[i] == 'o' &&line[i + 1] == 'n'))) {

j++; i++;

}

else if (!(line[i + 1] == 'n'&&line[i + 2] == 'g')) {

if ((line[i] == 'a' || line[i] == 'e') && (line[i + 1] == 'n' || line[i + 1] == 'r') && ((line[i + 2] == 'i') || (line[i + 2] == 'u') || (line[i + 2] == 'v') || (line[i + 2] == 'e'&&line[i + 3] == 'n'&&line[i + 4] == 'g') || (line[i + 2] == 'o'&&line[i + 3] == 'n')) || (line[i] == 'a'&&line[i + 1] == 'o'&&line[i + 2] == 'u')) {

j++; i++;

}

else {

i++; pin[j] += line[i]; j++; i++; //在这里改

}

}

else {

if (line[i] == 'a' && (line[i + 3] == 'u' || (line[i + 3] == 'e'&&line[i + 4] == 'n'&&line[i + 5] == 'g') || (line[i + 3] == 'o'&&line[i + 4] == 'n'))) {

i++; pin[j] += line[i]; j++; i++;

}

else {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

}

}

else if (line[i] == 't'&& line[i + 1] == 'x') {

pin[j] = "txushan";//初步处理，肯定不对，待会儿修改

j++; i = i + 7;

}

else if (line[i] == '\'') {

i++;

}

else {//假如第一个是声母

if (line[i + 1] == 'v') {

i++; pin[j] += line[i]; j++; i++;

}

else if (line[i + 1] == 'h' && (line[i] == 's' || line[i] == 'z' || line[i] == 'c')) {

pin[j] += line[i + 1]; s = 1; i++;

}

else if (!((line[i + 1] == 'a'&&line[i + 2] == 'i') || (line[i + 1] == 'a'&&line[i + 2] == 'o') || (line[i + 1] == 'a'&&line[i + 2] == 'n')

|| (line[i + 1] == 'e'&&line[i + 2] == 'i') || (line[i + 1] == 'e'&&line[i + 2] == 'n')

|| (line[i + 1] == 'o'&&line[i + 2] == 'u') || (line[i + 1] == 'o'&&line[i + 2] == 'n')

|| (line[i + 1] == 'i'&&line[i + 2] == 'a') || (line[i + 1] == 'i'&&line[i + 2] == 'e') || (line[i + 1] == 'i'&&line[i + 2] == 'o') || (line[i + 1] == 'i'&&line[i + 2] == 'u') || (line[i + 1] == 'i'&&line[i + 2] == 'n')

|| (line[i + 1] == 'u'&&line[i + 2] == 'a') || (line[i + 1] == 'u'&&line[i + 2] == 'e') || (line[i + 1] == 'u'&&line[i + 2] == 'o') || (line[i + 1] == 'u'&&line[i + 2] == 'i') || (line[i + 1] == 'u'&&line[i + 2] == 'n')

)) {

i++; pin[j] += line[i]; j++; i++;

}

else if (!((line[i + 1] == 'a'&&line[i + 2] == 'n'&&line[i + 3] == 'g') || (line[i + 1] == 'e'&&line[i + 2] == 'n'&&line[i + 3] == 'g') || (line[i + 1] == 'o'&&line[i + 2] == 'n'&&line[i + 3] == 'g')

|| (line[i + 1] == 'i'&&line[i + 2] == 'a'&&line[i + 3] == 'o') || (line[i + 1] == 'i'&&line[i + 2] == 'a'&&line[i + 3] == 'n') || (line[i + 1] == 'i'&&line[i + 2] == 'o'&&line[i + 3] == 'n') || (line[i + 1] == 'i'&&line[i + 2] == 'n'&&line[i + 3] == 'g')

|| (line[i + 1] == 'u'&&line[i + 2] == 'a'&&line[i + 3] == 'i') || (line[i + 1] == 'u'&&line[i + 2] == 'a'&&line[i + 3] == 'n')

)) {

if (((line[i + 1] == 'a' || line[i + 1] == 'e') && (line[i + 2] == 'n' || line[i + 2] == 'r') && ((line[i + 3] == 'i') || (line[i + 3] == 'u') || (line[i + 3] == 'v') || (line[i + 3] == 'e'&&line[i + 4] == 'n'&&line[i + 5] == 'g') || (line[i + 3] == 'o'&&line[i + 4] == 'n')))

|| (line[i + 1] == 'a'&&line[i + 2] == 'o'&&line[i + 3] == 'u')

|| ((line[i + 1] == 'u' || line[i + 1] == 'i') && line[i + 2] == 'e' && (line[i + 3] == 'n' || line[i + 3] == 'r') && ((line[i + 4] != 'a'&&line[i + 4] != 'e'&&line[i + 4] != 'i'&&line[i + 4] != 'o'&&line[i + 4] != 'u'&&line[i + 4] != 'v')))

|| ((line[i + 1] == 'i' || line[i + 1] == 'u') && line[i + 2] == 'e'&&line[i + 3] == 'i')

|| (line[i + 1] == 'i'&&line[i + 2] == 'a'&&line[i + 3] == 'i')

/\*绝对正确\*/ || ((line[i + 1] == 'i' || line[i + 1] == 'u') && line[i + 2] == 'n' && (line[i + 3] == 'i' || line[i + 3] == 'u' || line[i + 3] == 'v' || (line[i + 3] == 'e'&&line[i + 4] == 'n'&&line[i + 5] == 'g') || (line[i + 3] == 'o'&&line[i + 4] == 'n')))

|| (line[i + 1] == 'u'&&line[i + 2] == 'a'&&line[i + 3] == 'o'&&line[i + 4] != 'u')

) {

i++; pin[j] += line[i]; j++; i++;

}

else {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

}

else if (line[i + 4] != 'g'||(line[i+3]=='g'&&line[i+4]=='g')) {

if ((line[i + 1] == 'e' || line[i + 1] == 'a' || line[i + 1] == 'i') && line[i + 2] == 'n'&&line[i + 3] == 'g' && (line[i + 4] == 'u' || (line[i + 4] == 'e'&&line[i + 5] == 'n'&&line[i + 6] == 'g') || (line[i + 4] == 'o'&&line[i + 5] == 'n'))

|| (line[i + 1] == 'i'&&line[i + 2] == 'a'&&line[i + 3] == 'o'&&line[i + 4] == 'u')

|| ((line[i + 1] == 'i' || line[i + 1] == 'u') && line[i + 2] == 'a'&&line[i + 3] == 'n' && (line[i + 4] == 'u' || line[i + 4] == 'v' || line[i + 4] == 'i' || (line[i + 4] == 'e'&&line[i + 5] == 'n'&&line[i + 6] == 'g') || (line[i + 4] == 'o'&&line[i + 5] == 'n')))

) {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

else {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

}

else {

if ((line[i + 1] == 'i' || line[i + 1] == 'u') && line[i + 2] == 'a' && (line[i + 5] == 'u' || (line[i + 5] == 'e'&&line[i + 6] == 'n'&&line[i + 7] == 'g') || (line[i + 5] == 'o'&&line[i + 6] == 'n'))) {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

else {

i++; pin[j] += line[i]; i++; pin[j] += line[i]; i++; pin[j] += line[i]; i++; pin[j] += line[i]; j++; i++;

}

}

}

}

for (i = 0; i < j; i++) {

intpin[i].p = pin[i];

}

cout << "切分后的拼音有" << j << "个" << "是" << endl;

n = j;

for (i = 0; i < n; i++) {

for (j = 0; j < 403; j++)

if (Pinyin[j].p == intpin[i].p||intpin[i].p=="T恤衫") {

intpin[i] = Pinyin[j]; break;

}

if (j == 403)cout << "抱歉，没有找到对应的拼音" << endl;

}

}

**4.通过拼音串寻找可能的候选词的函数dealwithword**

**对于word中的每个词，假如它的第一个字与当前拼音切分的候选字中的某一个相同，那么对它的第二个字和下一个拼音切分进行判断。以此类推，假如这个词的所有字的拼音与已知拼音切分对应，那么这个词就是候选词，把它存入choice数组中。**

void dealwithword(word total[], int &nn1, int n, int nn)

{

int i, j, k;

for (i = 0; i < n; i++)

for (j = 0; j < total[i].sum&&j<nn; j++) {

for (k = 0; intpin[j].character[k].s != "0"; k++)

if (total[i].str[j] == intpin[j].character[k].s) break;

if (intpin[j].character[k].s == "0") break;

else if (j == total[i].sum - 1 && total[i].sum != 1) //就这句

choice[nn1++] = total[i];

}

}

**5.控制输出的函数show**

**先判断有没有词典，如果没有，建立词典，；如果有，进行builtuserword函数操作，读入用户词典中的词，形式与word.txt一样。**

void builtuserword(pinyin Pinyin[], word user[], int &n2)

{

ifstream in("userwords.txt");

if (!in)

{

cout << "打开文件失败" << endl;

}

word t;

char s[40];

int i, j, k, l;

n2 = 0;

string temp;

in >> s;//跳过"用户词典"

while (in >> s) {

j = 0; l = 0;

for (i = 0; s[i]; i = i + 2) {

temp = s[i]; temp += s[i + 1];

t.str[j] = temp; j++;

}

t.sum = j;

user[n2++] = t;

}

}

读入后先对word.txt中的词调用dealwithword找出候选词，再对用户词典中的词调用dealwithword找出候选词，

输出时，把choice中的候选词倒序输出，这样，用户词典中的候选词就会显示在前面，之后再显示当前拼音所对应的候选字。用户进行选择后把当前选的字或词存入string类型变量s中，拼音移到下一个，循环。

当所有拼音都选择完后，在屏幕上输出最终结果（s），并把s输出到用户词典中。

void show(int &nn1, int &n)

{

int i, j, x, k, t;

t = 0;

string s;

fstream file;

file.open("userwords.txt", ios::in);

if (!file) {

cout << "目前没有词典" << endl;

}

else {

t = 1;

cout << "已经有词典了！" << endl;

builtuserword(Pinyin, user, n2);

}

ofstream out;

out.open("userwords.txt");

out << "用户词典" << endl;

if (t == 1) {

for (i = 0; i < n2; i++) {

for (j = 0; j < user[i].sum; j++)

out << user[i].str[j];

out << endl;

}

}

for (; ;) {

nn1 = 0;

cout << "候选词序列：" << endl;

if (intpin[0].p == "txushan") {

for (i = 0; i < n - 1; i++)

intpin[i] = intpin[i + 1];

n = n - 1;

cout << 1 << "T恤衫" << endl;

s += "T恤衫";

if (n == 0) {

cout << "最终结果" << endl;

cout << s << endl;

break;

}

else continue;

}

dealwithword(total, nn1, n1, n);

if (t == 1) dealwithword(user, nn1, n2, n);

for (i = nn1 - 1; i >= 0; i--) {

cout << nn1 - i;

for (j = 0; j < choice[i].sum; j++)

cout << choice[i].str[j];

cout << '\t';

}

for (i = nn1; intpin[0].character[i - nn1].s != "0"; i++)

cout << 1 + i << intpin[0].character[i - nn1].s << '\t';

cout << endl;

cout << "请输入要选择的词的序号" << endl;

cin >> x;

if (x <= nn1) {

for (j = 0; j < choice[nn1 - x].sum; j++)

s += choice[nn1 - x].str[j];

k = choice[nn1 - x].sum;

}

else {

s += intpin[0].character[x - nn1 - 1].s;

k = 1;

}

for (i = 0; i < n - k; i++)

intpin[i] = intpin[i + k];

n = n - k;

if (n == 0) {

cout << "最终结果:" << endl;

cout << s << endl;

out << s << endl;

break;

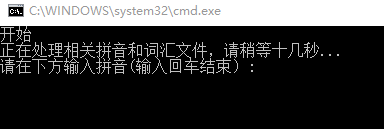
}

}

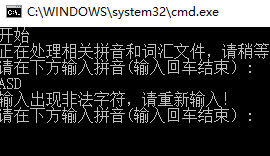
}

**五、程序运行截图**

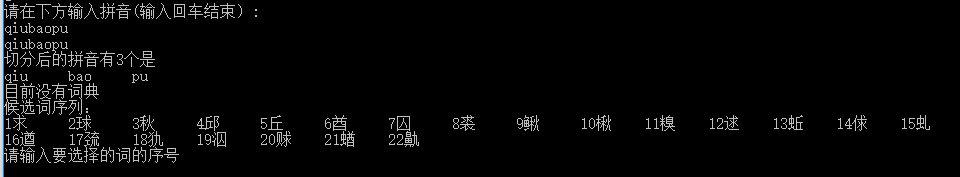
1.开始



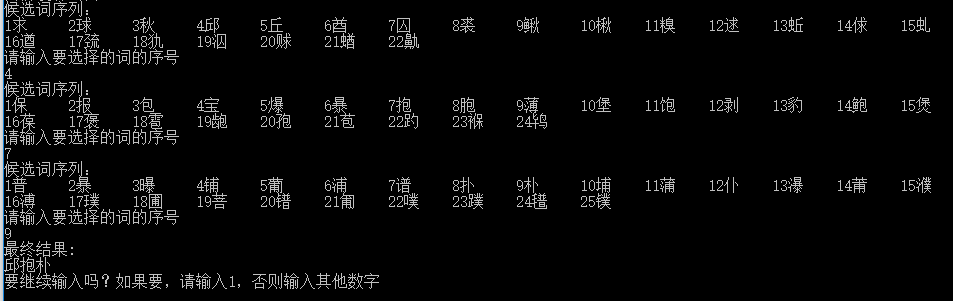
2.如果输入非法字符



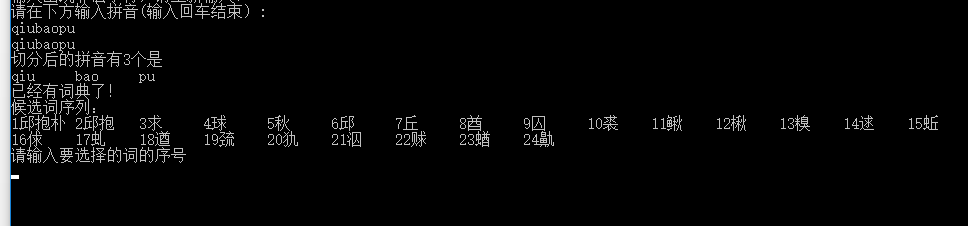
3.正确输入后，输出拼音划分，并按照用户词典——候选词——候选字的顺序输出候选词序列，供用户选择



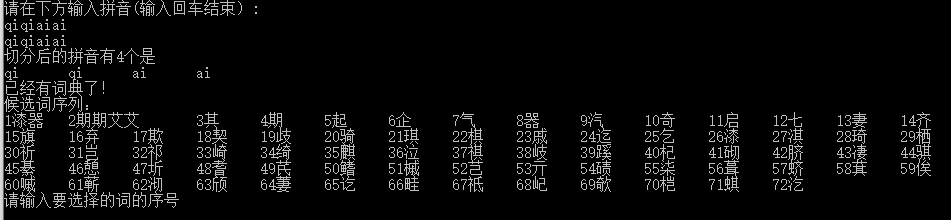
4.选择完成后，显示最终结果，并存入用户词典



5.下一次再输入相同的拼音序列，优先显示用户之前选过的词



6.对于某些不应按照正向最大匹配的拼音（qiqiaiai），该程序也可以正确划分。



7.输入1继续输入，输入其他数字退出程序

