阅读下面这段代码，说出它的目的，每个成员函数的作用，最终的输出结果以及设计思路：

#include <stdio.h>

#include <string.h>

class PyNetMaker

{

public:

void MainProc(const char \*inputStr);

protected:

short\*\* preProcess(const char\* inputStr);

void printPyNet(const char\* inputStr, int\* route, int top, short\*\* arcArray);

const char\* mPyMap[] = {"a", "ai", "an", "ang", ...

"gu", "gua", "guai", ..., "guo", ...

"o", "ou", ...

"xi", "xia", "xiang", ...

};// 共412个合法的音节串，按字母序排列

};

short\*\* PyNetMaker::preProcess(const char\* inputStr)

{

if(inputStr == NULL || strlen(inputStr) == 0)

return NULL;

short\*\* arcArray = (short\*\*)malloc(sizeof(short\*) \* strlen(inputStr));

for(int i=0; i<strlen(inputStr); i++){

const char\* tmpStr = inputStr + i;

static const int nMaxArc = 16;

short arcLen[nMaxArc] = {0};

int arcCount = 0;

for(int j=0; j<(sizeof(mPyMap) / sizeof(char\*)); j++){

char\* syllable = mPyMap[j];

if(syllable[0] > tmpStr[0])

break;

if(strstr(tmpStr, syllable) == tmpStr){

arcLen[arcCount++] = strlen(syllable) - 1;

}

}

arcArray[i] = (short\*)malloc(sizeof(short) \* arcCount + 1);

arcArray[i][0] = arcCount;

memcpy(arcArray[i] + 1, arcLen, sizeof(short) \* arcCount);

}

return arcArray;

}

void PyNetMaker::printPyNet(const char\* inputStr, int\* route, int top, short\*\* arcArray)

{

char result[256] = {0};

for(int i=0; i<top; i++){

int value = route[i];

short nStart = (value >> 16) & 0x0000ffff;

short nIdx = value & 0x0000ffff;

strncat(result, inputStr + nStart, arcArray[nStart][nIdx + 1] + 1);

strcat(result, "'");

}

printf("%s\n", result);

}

void PyNetMaker::MainProc(const char\* inputStr)

{

if(inputStr == NULL || strlen(inputStr) == 0)

return;

short\*\* arcArray = preProcess(inputStr);

int nStart = 0;

int nIdx = 0;

int\* route = (int\*)malloc(sizeof(int) \* strlen(inputStr));

int top = 0;

while(1){

if(nStart<strlen(inputStr) && arcArray[nStart][0]!=0 && nIdx<arcArray[nStart][0]){

route[top++] = (((nStart << 16) & 0xffff0000) | (nIdx & 0x0000ffff));

nStart += arcArray[nStart][nIdx + 1] + 1;

nIdx = 0;

if(nStart == strlen(inputStr))

printPyNet(inputStr, route, top, arcArray);

}else{

if(nStart == 0)

return;

int value = route[--top];

nStart = (value >> 16) & 0x0000ffff;

nIdx = (value & 0x0000ffff) + 1;

}

}

free(route);

for(int i=0; i<strlen(inputStr); i++)

free(arcArray[i]);

free(arcArray);

}

int main(int argc, char \*argv[])

{

PyNetMaker pyNetMaker();

pyNetMaker.MainProc("xianguo");

return 0;

}