/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LeetCode 273 难度：困难

题目：将非负整数转换为其对应的英文表示。可以保证给定输入小于 231 - 1 。

示例 1:

输入: 123

输出 : "One Hundred Twenty Three"

示例 2 :

输入 : 12345

输出 : "Twelve Thousand Three Hundred Forty Five"

示例 3 :

输入 : 1234567

输出 : "One Million Two Hundred Thirty Four Thousand Five Hundred Sixty Seven"

示例 4 :

输入 : 1234567891

输出 : "One Billion Two Hundred Thirty Four Million Five Hundred Sixty Seven Thousand Eight Hundred Ninety One"

编译环境：Visual Studio 2015

作者：杨政权

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//具体代码

#include "stdafx.h"

#include "iostream"

#include <vector>

#include <string>

#include <stdlib.h>

using namespace std;

//算法

class Solution {

public:

//三个数一划分 依次为：Thousand Million Billion

string numberToWords(unsigned int num) {

string res = convertHundred(num % 1000);

vector<string> v = { "Thousand", "Million", "Billion" };

for (int i = 0; i < 3; ++i) {

num /= 1000; //取整

res = num % 1000 ?( convertHundred(num % 1000) + " " + v[i] + " " + res ): res;

//三目运算符，取余

}

while (res.back() == ' ') res.pop\_back(); //满足条件，弹出

return res.empty() ? "Zero" : res; //空返回0，否则返回res

}

//三个数的读法

string convertHundred(int num) {

//后两位小于20的时候，有具体的单词

vector<string> v1 = { "", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };

//大于20时，进行拼接

vector<string> v2 = { "", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };

string res;

//取出高位 后两位 最低位数字

int a = num / 100, b = num % 100, c = num % 10;

res = b < 20 ?( v1[b]) : (v2[b / 10] + (c ? " " + v1[c] : ""));

if (a > 0) res = v1[a] + " Hundred" + (b ? " " + res : "");

return res;

}

};

void main() {

string NumberToString;

Solution s;

unsigned int number;

//提示用户输入数字

cout << "请输入一个数字:" << endl;

cin >> number;

//NumberToString接收函数返回值

NumberToString = s.numberToWords(number);

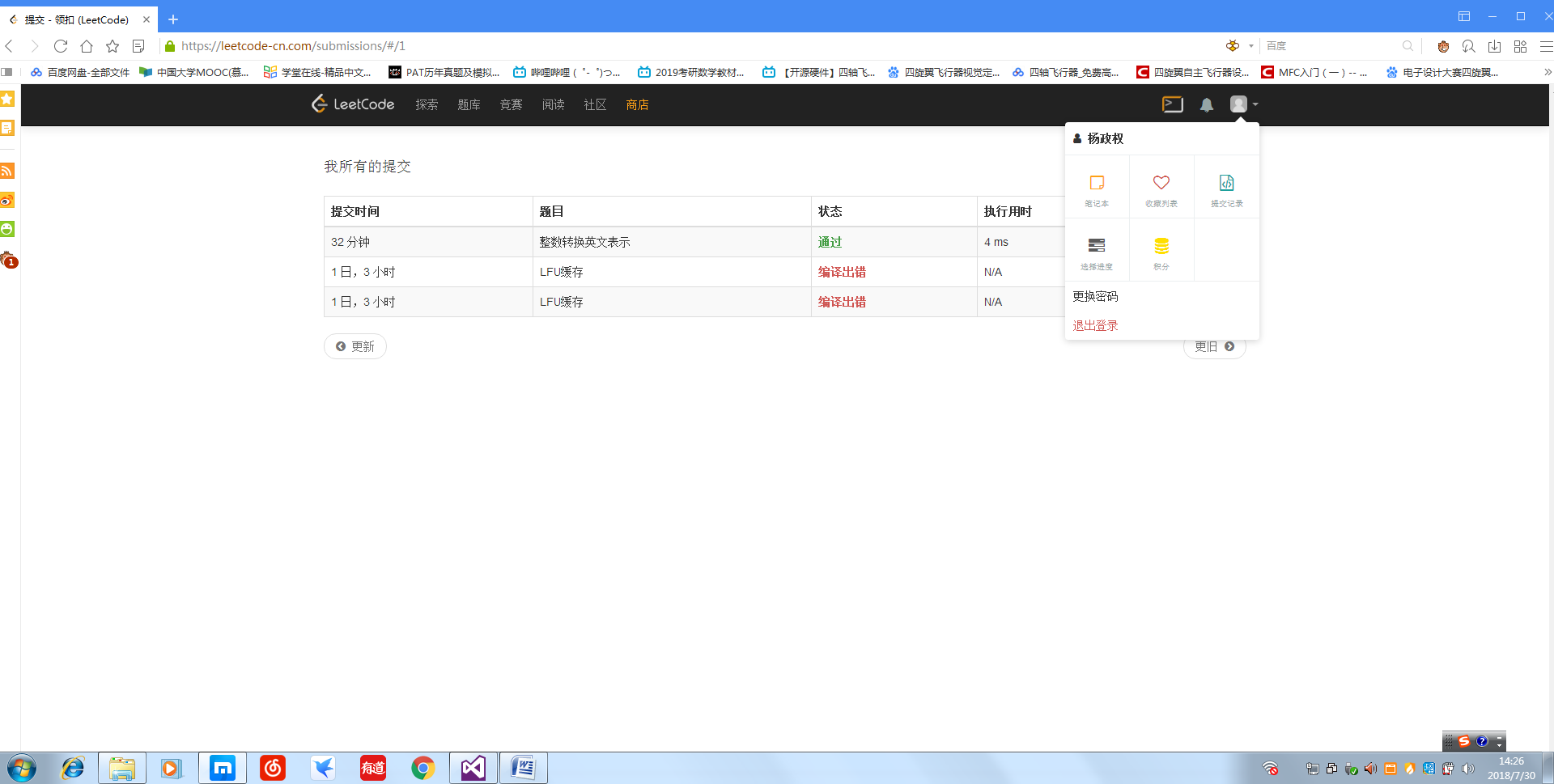
//将结果打印在屏幕上输出

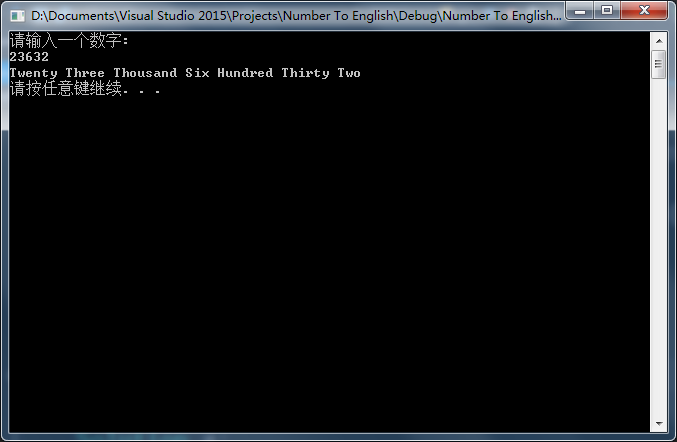
cout << NumberToString << endl;

system("pause");

}

//LeetCode运行成功截图以及VS2015上面的截图



//实验总结：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* LeetCode上面归为困难

\*\*\* 但感觉也不是很难

\*\*\* 主要是数字小于20时有具体的单词所表示

\*\*\* 这点不容易想到，被常规思维所局限

\*\*\* 还有就是建立数组时要考虑后面的算法

\*\*\* 否则会吃大亏

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/