# 题目

中心对称数是指一个数字在旋转了 180 度之后看起来依旧相同的数字（或者上下颠倒地看）。

请写一个函数来判断该数字是否是中心对称数，其输入将会以一个字符串的形式来表达数字。

示例 1:

输入: num = "69"

输出: true

示例 2:

输入: num = "88"

输出: true

示例 3:

输入: num = "962"

输出: false

示例 4：

输入：num = "1"

输出：true

# 分析

class Solution

{

public:

bool isStrobogrammatic(string num)

{

unordered\_map<char,char> dic = {{'0','0'}, {'1','1'}, {'6','9'}, {'8','8'}, {'9','6'}};

string tmp;

for (char c: num)

{

if (dic.count(c))

{

tmp += dic[c];

}

else //其他的数字，不可能

return false;

}

reverse(tmp.begin(), tmp.end());

return tmp == num;

}

};

当数字中有2，3，4，5，7的时候，直接返回false;  
 之后从字符串num结尾开始遍历，遇到的数字都倒过来;  
 如果最后返回str是否等于num;

class Solution {

public:

bool isStrobogrammatic(string num) {

char shuzu[]={'2','3','4','5','7'};

for(int i=0;i<num.size();i++){

for(auto &x:shuzu){

if(num[i]==x) return false;

}

}

string str="";

for(int i=num.size()-1;i>=0;i--){

if(num[i]=='1') str+='1';

else if(num[i]=='6') str+='9';

else if(num[i]=='8') str+='8';

else if(num[i]=='9') str+='6';

else if(num[i]=='0') str+='0';

}

return str==num;

}

};

class Solution {

public:

unordered\_map<char, char> M = {

{'1', '1'}, {'0', '0'}, {'8', '8'},

{'6', '9'}, {'9', '6'}};

bool isStrobogrammatic(string num) {

int N = num.size();

for (int i = 0; i <= N / 2; ++i) {

if (M.count(num[i]) == 0 || M[num[i]] != num[N - 1 - i]) return false;

}

return true;

}

};