strobogrammatic ii

A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Find all strobogrammatic numbers that are of length = n.

For example,

Given n = 2, return ["11","69","88","96"].

```
#include<iostream>
#include<map>
#include<algorithm>
#include<vector>
#include<string>
using namespace std;
vector<string> findStrobogrammatic(int n)
{
       string ss[] = {"0", "1", "8"};
       string bases[] = {"00", "11", "88", "69", "96"};
       vector <string> base;
       vector<string> strobos;
    if (n & 1)
    {
       for(int k=0;k<3;k++) strobos.push_back(ss[k]);
    }else{
       strobos.push_back("");
    }
    for(int k=0;k<5;k++) base.push_back(bases[k]);
    int m = base.size();
    while(n>1)
    {
       n-=2;
       vector<string> temp;
       for(int k=0;k<strobos.size();k++)
       {
              int i=n<2?1:0;
              // mark(zzw) if the size of n less than 2
              // i.e., strobos[k] = 'xxx' 101 609 ...
              // then we can not add 00 at the beginning or end
              // e.g., 010 080 is not a number so that n=1
              for(;i<m;i++)
              {
                     temp.push_back(base[i].substr(0,1)+strobos[k]+base[i].substr(1));
              }
          }
          swap(temp,strobos);
    }
    return strobos;
}

int main(int argc,char *argv[])
{
       vector<string> ans = findStrobogrammatic(3);
       for(int i=0;i<ans.size();++i)
              cout<< ans[i]<< endl;
       return 0;
}
```

strobogrammatic iii

A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Write a function to count the total strobogrammatic numbers that exist in the range of low <= num <= high.

For example,

Given low = "50", high = "100", return 3. Because 69, 88, and 96 are three strobogrammatic numbers.

```
bool less2(string &s,string &t)
{
      int m = s.length(), n = t.length(), i;
    if (m != n) return m < n;
    for (i = 0; i < m; i++)
    {
       if (s[i] == t[i]) continue;
       else break;
    }
    return i == m || s[i] < t[i];
}

void strobogrammaticCount(string temp,int &ans,string &low,
      string &high,int lo,int hi)
{
      if(lo>hi)
      {
            // mark(zzw):
            // 1.the first number of temp!=0 i.e., 010 080 is illegal
            // 2.temp>0 && temp<9 i.e., 0,1,2,3...9
            if((temp[0]!='0' || temp.length()==1) && less2(low,temp) &&
less2(temp,high))
            {
                  ans++;
                  return;
            }
      }
      map<char,char>::iterator it;
      for(it=mm.begin();it!=mm.end();it++)
      {
            temp[lo] = it->first;
            temp[hi] = it->second;
            if((lo==hi && it->first==it->second) || lo<hi)
            {
                  strobogrammaticCount(temp,ans,low,high,lo+1,hi-1);
            }
      }
}

int strobogrammaticInRange(string low,string high)
{
      int l = low.length();
      int u = high.length();
      int ans = 0;
      for(int i=l;i<=u;i++)
      {
            string temp = string(i,' ');
            strobogrammaticCount(temp,ans,low,high,0,i-1);
      }
      return ans;
}
```

```
int main(int argc,char *argv[])
{
	mm.insert(make_pair('0','0'));
	mm.insert(make_pair('1','1'));
	mm.insert(make_pair('8','8'));
	mm.insert(make_pair('6','9'));
	mm.insert(make_pair('9','6'));
	map<char,char>::iterator it;
	int ans2 = strobogrammaticInRange("50","100");
	cout << ans2 << endl;
	mm.clear();
	return 0;
}
```