/\*

Find all possible combinations of k numbers that add up to a number n,

given that only numbers from 1 to 9 can be used and each combination should be a unique set of numbers.

Example 1

Input: k = 3, n = 7

Output:[[1,2,4]]

Example 2:

Input: k = 3, n = 9

Output:[[1,2,6], [1,3,5], [2,3,4]]

题目：找1-9中k个数凑成n

思路：由于k不定，只能backtracking

\*/

class Solution {

public:

void combine(vector<vector<int>> & ret,vector<int> m1,int target,int i,int k)

{

if(target>k\*9 || k==0)

return;

if(k==1 && target>=i)

{

m1.push\_back(target);

ret.push\_back(m1);

return;

}

else if(k==1 && target<i)

return;

else //k>1 k代表还剩几个数来凑

{

for(int j=i;j<=9;j++)

{

m1.push\_back(j);

combine(ret,m1,target-j,j+1,k-1);

m1.pop\_back();

}

}

}

vector<vector<int>> combinationSum3(int k, int n)

{

vector<vector<int>> ret;

if(n>k\*9)

return ret;

vector<int> m1;

combine(ret,m1,n,1,k);

return ret;

}

};