```cpp
/***
A number is said to be a 369 number if The count of 3s is equal to count of
6s and the count of 6s is equal to count of 9s. The count of 3s is at least 1.
For Example 12369,383676989,396 all are 369 numbers whereas 213,342143,111 are not.
Given A and B find how many 369 numbers are there in the interval [A, B].
Print the answer modulo 1000000007.

The first line contains the number of test cases (T) followed by T lines
each containing 2 integers A and B. T<=100 and 1<=A<=B<=10^50.

For each test case output the number of 369 numbers between A and B inclusive.

Input               Output
3
121 4325       -----> 60
432 4356       -----> 58
4234 4325667   -----> 207159

***/
#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007LL
#define ll long long
int dp[51][2][2][17][17][17];
string LB,UB;
int fun(int pos, int boro, int choto, int three, int six, int nine){
    if(three>16 || six>16 || nine>16)return 0LL;
    if(pos==51){
        if(three>0 && three==six && six==nine) return 1LL;
        return 0LL;
    }

    int &ret = dp[pos][boro][choto][three][six][nine];
    if(ret!=-1 && choto && boro) return ret;

    int lo = 0, hi=9;
    if(boro==0) lo = LB[pos]-'0';
    if(choto==0)hi = UB[pos]-'0';
    ret = 0;
    for(int i=lo; i<=hi; i++){
        ret += fun(pos+1, boro|(i>lo), choto|(i<hi), three+(i==3), six+(i==6), nine+(i==9));
        ret %= MOD;
    }
    return ret;
}
int main(){
    memset(dp,-1,sizeof(dp));
    int tt; scanf("%d",&tt);
    for(int ks=1; ks<=tt; ks++){
        cin >> LB;
        string p;
        int d = 51-LB.size();
        for(int i=1; i<=d; i++)p+="0";
        LB = p+LB;

        cin >> UB;
        string q;
        int c = 51-UB.size();
        for(int i=1; i<=c; i++)q+="0";
        UB = q+UB;

        int ans = fun(0,0,0,0,0,0);
        cout << ans << endl;
    }
    return 0;
}
```