**Problem Name – Playing With Digits**

**Problem Statement:**

You are given four values *L, R, X* and *Y*. Count the number of integers in the range *[L, R]* with the following property :

Number of digits with Odd Frequency = *X* and number of digits with Even Frequency = *Y*.

Also (*X + Y = 10*)

Here, frequency of digit in an integer is count of the number of times it occurs in decimal representation of numbers (without leading zeros).

**Note :** If a digit '*d*' occurs in a number 0 times, then it is considered to have even frequency.

As the answer could be large, find it modulo (109+ 7).

**Input Format:**

* The first line will contain four values *L, R, X, and Y*.

**Output Format:**

* Number of integers in *[L,R]* with given property modulo (10⁹ + 7).

**Constraints:**

* 1 ≤ *L* ≤ *R* ≤ 10150
* 0 ≤ *X,Y* ≤ 10
* *X + Y* = 10.

**Sample Input 1:**

10 20 2 8

**Sample Output 1:**

10

**Explanation:**

There are 10 integers satisfying the given condition from range [10,20] are -

10, 12, 13, 14, 15, 16, 17, 18, 19, 20.

**Sample Input 2:**

1 1000 10 0

**Sample Output 2:**

9

**Solution:**

#include <bits/stdc++.h>

#define int long long

#define sz(x) (int)(x.size())

using namespace std;

const int N = 155, mod = 1e9 + 7, inf = 1e18;

string A, B;

int n, dp[N][2][2][2][1 << 10];

int even, odd;

int go(int pos, bool bigA, bool smallB, int freq\_mask, int nonZeroTaken){

if(pos >= n){

int oddCnt = \_\_builtin\_popcountll(freq\_mask);

int evenCnt = 10 - oddCnt;

return oddCnt == odd && evenCnt == even;

}

int &ans = dp[pos][bigA][smallB][nonZeroTaken][freq\_mask];

if(~ans)

return ans;

ans = 0;

for(int i = 0; i < 10; i++){

if(!bigA && i < A[pos] - '0') continue;

if(!smallB && i > B[pos] - '0') continue;

int next\_freq\_mask = freq\_mask;

if(nonZeroTaken || (i != 0)){

next\_freq\_mask ^= (1 << i);

}

ans = (ans + go(pos + 1, bigA || (i > A[pos] - '0'), smallB || (i < B[pos] - '0'), next\_freq\_mask, nonZeroTaken || (i != 0))) % mod;

}

return ans;

}

int32\_t main(){

ios::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

memset(dp, -1, sizeof(dp));

cin >> A >> B;

cin >> odd>> even;

assert(even + odd == 10);

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

while(sz(A) < sz(B))

A.push\_back('0');

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

n = sz(B);

int ans = 0;

for(int i = 0; i < 10; i++){

if(i < A[0] - '0') continue;

if(i > B[0] - '0') continue;

int freq\_mask = 0;

if(i != 0)

freq\_mask ^= (1 << i);

ans = (ans + go(1, i > A[0] - '0', i < B[0] - '0', freq\_mask, i != 0)) % mod;

}

cout << ans << '\n';

}