数位DP：

typedef long long ll;

int a[20];

ll dp[20][st];//状态

ll dfs(int p,int st,bool zero,bool limit)//不是每个题都要判断前导零

{

if(!p) return 1; /\*一般返回1,视情况而定 \*/

if(!limit && !lead && dp[p][st]!=-1) return dp[p][st];

int up=limit?a[p]:9;

ll tmp=0; //计数

for(int i=0;i<=up;i++)

{

nst=t,…;

if() ...

if()…

tmp+=dfs(p-1,nst,zero && i==0,limit && i==a[p]);

}

if(!limit && !zero) dp[p][st]=tmp;

return ans;

}

ll solve(ll x)

{

int p=0;

while(x)

{

a[++p]=x%10;/

x/=10;

}

return dfs(p-1,st,1,1);//刚开始都是有限制且有前导零

}

int main()

{

ll l,r;

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

while(~scanf("%lld%lld",&l,&r))

{

printf("%lld\n",solve(r)-solve(l-1));

}

}