```cpp
#include<cstring>
#include<cstdio>
#include<iostream>
#include<algorithm>
#include<vector>
#include<cmath>
#include<map>
#define l(x) (x<<1)
#define r(x) ((x<<1)+1)
#define IL inline
#define reg register
#define LL long long
#define INF 0x3f3f3f3f
using namespace std;

int i,len,mod;
LL a,b;
int z[20];
LL mi[19],f[20][200][200];

IL int Abs(int x){return (x<0)?-x:x;}
IL void Swap(int &a,int &b){a^=b^=a^=b;}
IL int Min(int a,int b){return (a<b)?a:b;}
IL int Max(int a,int b){return (a>b)?a:b;}

IL LL read(){
    LL p=0,f=1; char   c=getchar();
    while (c<48||c>57)  {if (c=='-')   f=-1;   c=getchar();}
    while (c>=48&&c<=57)   p=(p<<1)+(p<<3)+c-48,c=getchar();
    return p*f;
}

inline LL Mi(LL x,LL y,int MOD){
    LL p=x,t=1,Res=1;
    for (;t<=y;(t&y)?Res=(Res*p)%MOD:0,p=(p*p)%MOD,t<<=1);
    return Res;
}

IL LL Dfs(int pos,int lead,int limit,LL tot,LL Sum){
    reg LL i=0,up=0,sum=0,res=0;
    if (pos>len){
        if (Sum!=mod) return 0;
        return (!(tot%mod))?1:0;
    }
    res=(tot*(mi[len-pos+1]%mod))%mod;
    if (f[pos][res][Sum]!=-1 && !limit && !lead)    return f[pos][res][Sum];
    up=(limit)?z[len-pos+1]:9;
    for (i=0;i<=up;i++){
        if (!i && lead) sum+=Dfs(pos+1,1,limit&&(i==up),tot,Sum);
        else if (i && lead)
sum+=Dfs(pos+1,0,limit&&(i==up),(tot<<3)+(tot<<1)+i,Sum+i);
```

```cpp
        else sum+=Dfs(pos+1,0,limit&&(i==up),(tot<<3)+(tot<<1)+i,Sum+i);
    }
    return (!limit && !lead)?f[pos][res][Sum]=sum:sum;
}

IL LL Cal(LL x){
    reg LL i=0,ans=0;
    len=0;
    while (x){z[++len]=x%10;    x/=10;}
    for (i=1;i<=9*len;i++){
        memset(f,-1,sizeof(f));
        mod=i;
        ans+=Dfs(1,1,1,0,0);
    }
    return ans;
}

int main(){
    #ifdef __Marvolo
    freopen("zht.in","r",stdin);
    freopen("zht.out","w",stdout);
    #endif
    a=read();    b=read();
    for (mi[1]=10,i=2;i<=18;i++)    mi[i]=mi[i-1]*10;
    cout<<Cal(b)-Cal(a-1)<<endl;
    return 0;
}
```