DIGIT DP :

From L to R how many numbers are divisible by K & sum of their digits are also divisible by K.

```
#include<bits/stdc++.h>
using namespace std;
int dp[10][2][82][82],x[12],cnt,k;

void func(int num)
{
   cnt=0;
   while(num>0)
   {
      x[cnt++]=num%10;
      num/=10;
   }
   reverse(x,x+cnt);
}

int call(int i,int top,int digmod,int nummod)
{
   if(i==cnt)
      return (digmod==0)&&(nummod==0);
   if(dp[i][top][digmod][nummod]!=-1)
      return dp[i][top][digmod][nummod];
   int mx,res=0;
   if(top)
      mx=x[i];
   else
      mx=9;
   for(int j=0;j<=mx;j++)
   {
      res+=call(i+1,top&&(j==mx),(digmod+j)%k,((nummod*10)%k+j)%k);
   }
   return dp[i][top][digmod][nummod]=res;

}
```

```
int main()
{
   int t,cs=0;
   cin >> t;
   while(t--)
   {

      int a,b;
      scanf("%d%d%d",&a,&b,&k);
      a--;
      printf("Case %d: ",++cs);
      if(k>81)
         printf("0\n");
      else{
      memset(dp,-1,sizeof(dp));
      func(b);
      int ans=call(0,1,0,0);
      memset(dp,-1,sizeof(dp));
      func(a);
      ans-=call(0,1,0,0);
      printf("%d\n",ans);
      }

   }
}
```