

## LIGHTOJ-1140 - How Many Zeroes?(DIGIT DP):

How many zeroes in the decimal representations of all natural numbers between and including **m** and **n**, (**m ≤ n**) . Each case contains two unsigned 32-bit integers **m** and **n**, (**m ≤ n**) .

```
#include<bits/stdc++.h>
using namespace std;
long long dp[12][2][2][25],len;
char s[12];
long long call(int pos, int choto, int flag,long long zero)
{
    if(dp[pos][choto][flag][zero]!=-1)
        return dp[pos][choto][flag][zero];

    long long ret=0, next_zero, next_choto, next_flag;;
    if(pos==len) return zero;

    int low=0, high=9;
    if(choto==0) high=s[pos]-'0';

    for(int i=low; i<=high; i++)
    {
        if(choto==0){
            if(i==high) next_choto=0;
            else next_choto=1;
        }
        else
            next_choto=1;

        next_flag=flag;
        if(i>0) next_flag=1;

        next_zero=zero;
        if(flag && i==0) next_zero=zero+1;

        ret += call(pos+1,next_choto,next_flag,next_zero);
    }
    return dp[pos][choto][flag][zero]=ret;
}
```

```
int main()
{
    int T,test;
    long long cnt1,cnt2,n,m,res;

    cin >> T;

    for(test=1; test<=T; test++)
    {
        cin >> m >> n;
        if(m==0 && n==0) res=1;
        else
        {
            if(m!=0)
            {
                sprintf(s,"%lld",m-1);
                len=strlen(s);
                memset(dp,-1,sizeof(dp));
                cnt1 = call(0,0,0,0);
            }
            else
                cnt1=-1;

            sprintf(s,"%lld",n);
            len=strlen(s);
            memset(dp,-1,sizeof(dp));
            cnt2 = call(0,0,0,0);

            res=abs(cnt2-cnt1);
        }
        printf("Case %d: %lld\n",test,res);
    }
    return 0;
}
```