// 逆元 /b=\*b’ (a/b)%c = (a\*b’)%c 其中b’为b的逆元..

#include<bits/stdc++.h>

typedef long long ll;

typedef unsigned long long ull;

#define IO ios\_base::sync\_with\_stdio(0),cin.tie(0)

using namespace std;

#define rep(i,a,b) for(ll i = a; i<=b ;++i)

#define per(i,a,b) for(ll i = a; i>=b ;--i)

#define maxn 30

#define mod 1000000007

int c[maxn][maxn];

string s;

ll k, tmp, n, ans = 0, x0;

void exgcd(ll a, ll b, ll &d, ll &x, ll &y)

{

if(!b)

{d = a; x = 1; y = 0;

}

else

{

exgcd(b, a%b, d, y, x);

y -= a/b \* x;

}

}

///(a/b)%c这类运算不能等价于(a%c / b%c)―-可等价为(a\*b’)%c…其中b’为b的逆元..

ll inverse(ll a, ll n)//求逆元

{

ll x, y, d;

exgcd(a, n, d, x, y);

if( d == 1)

return (x + mod) %mod;

else return -1;

}

ll poww(ll a, ll b, ll p)

{

ll ans = 1, base = a;

while( b )

{

if( b&1 )

{

ans \*= base;

ans = ans % p;

}

base = (base%p\*base%p)%p;

b >>= 1 %p;

}

return ans;

}

int main()

{

cin >> s >> k;

tmp = 1, n = s.size();

rep(i, 0, n-1)

{

if( s[i] == '0' || s[i] == '5')

ans = (tmp + ans) %mod;

tmp = ( 2 \* tmp ) % mod;

///tmp 就是 公比 q

//cout << ans << ' ' << tmp << endl;

}

//每次都找0 5结尾的数，它前面的可以选或者不选就是总共2^i种可能.

//ans 种选择

ll an = poww(tmp, k, mod);

ll bb = inverse(tmp - 1 , mod);

//cout << 142857144 \* 7 - 1<< endl;

///等比数列求和 a1 \* (q的k次方-1)/(q-1)

// a1 \* (tmp的k次方-1)/(tmp-1)

///因为分子分母都可能会爆炸，又存在对除数取模的运算，这时就需要用到快速幂和逆元

//（2^(k+len)）,（2^(k+len\*2)）...,（2^(k+len\*(k-1)）

//2^n(2^(k\*len))/(2^len-1)）

cout << ans << ' ' << an << ' ' <<tmp << ' ' << bb << endl;

///t = ans \* (an-1) / ( tmp-1) )

ll t = ((( ans \* (an-1))%mod \* bb )%mod + mod) %mod;

cout << t;

}