思路:

1. 当 n 为奇数时, f[n] = f[n-1], 因为只需在所有的序列前添加一个 1 即可, 所有的序列同时延迟 1 位, 不会出现重复

　　若是这个 1 和其他的1组成 2 而不是放在首位, 怎么办? 不会这样, 因为这个序列肯定已经存在了

　　证明, 假设sum(s1) = 2\*k, s1内部某个1加1得到 s2, 则 sum(s2) = 2\*k+1, s2 的首位仍然肯定是1, 那么 s2 也可以通过 s3 延长而来, 所以必然已经存在了

2. 当 n 为偶数时, 分为两种情况

　　<1> 某个序列首位为1, 则该序列由 f(n-1) 延长而来

　　<2> 当某个序列首位为2, 则该序列没有1, 将该序列的所有元素除以 2, 则 是 f(n/2)的序列

      f[n] = f[n-1]+f[n/2]

求把一个整数分解为2的幂的和共有几种方案

6=1+1+1+1+1+1

6=1+1+1+1+2

6=1+1+2+2

6=1+1+4

6=2+2+2

6=2+4

7=1+1+1+1+1+1+1

7=1+1+1+1+1+2

7=1+1+1+2+2

7=1+1+1+4

7=1+2+2+2

7=1+2+4

如果i为奇数，肯定有一个1，把f[i-1]的每一种情况加一个1就得到fi,所以f[i]=f[i-1]

如果i为偶数，如果有1，至少有两个，则f[i-2]的每一种情况加两个1，就得到i，如果没有1，则把分解式中的每一项除2，则得到f[i/2]

所以f[i]=f[i-2]+f[i/2]

由于只要输出最后9个数位，别忘记模1000000000

#include <iostream>

#include <iomanip>

#include <stdio.h>

#include <stdlib.h>

#include <algorithm>

#include <functional>

#include <vector>

#include <cmath>

#include <string>

#include <stack>

#include<cstring>

#include <queue>

#define MOD 1000000000

using namespace std;

int dp[1000010];

int main()

{

// freopen("input.txt","r",stdin);

int n;

cin>>n;

dp[1]=1;

for(int i=2;i<=n;i++)

if(i&1)

dp[i]=dp[i-1];

else

dp[i]=(dp[i-1]+dp[i>>1])%MOD;

cout<<dp[n]<<"\n";

return 0;

}