**Coin Toss**

Bruce is very sad after being informed that Harvey is now killing people based on the outcome of a coin toss. He also knows that Harvey kills someone only when he remembers Rachel after tossing the coins.

Harvey tosses the coin finite number of times and lists the outcomes. In the list if Tails comes three times in a row, He remembers Rachel and kills the other person.

Bruce is sad but wants to focus on positives. so he thinks about the number of ways when the other person is not killed for given number of tosses. As he is unable to compute the desired number he asks you to do so.

Input:

First line of the Input contains an integer *T* denoting the number of testcases. Each testcase contains only one integer N denoting the number of times Harvey tosses the coin.

Output:

print a single integer denoting the answer%1000000007.

Constraints :

1<= T <= 10^5

1<= N <= 10^18

**SAMPLE INPUT**

2

4

5

**SAMPLE OUTPUT**

13

24

#include<bits/stdc++.h>

#define pii pair<int,int>

#define pb push\_back

#define INF 10000000

#define NINF -1000000000

#define mod 1000000007

#define int long long int

using namespace std;

void multiply(int A[][3],int B[][3])

{

int X[3][3];

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

X[i][j]=(((A[i][0])%mod\*(B[0][j])%mod)%mod+((A[i][1])%mod\*(B[1][j])%mod)%mod+((A[i][2])%mod\*(B[2][j])%mod)%mod)%mod;

}

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

A[i][j]=X[i][j];

}

}

void power(int A[][3],int n)

{

int res[3][3]={1,0,0,0,1,0,0,0,1};

while(n>0)

{

if(n&1)

multiply(res,A);

multiply(A,A);

n>>=1;

}

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

A[i][j]=res[i][j];

}

}

int ans(int n)

{

int A[3][3]={1,1,1,1,0,0,0,1,0};

power(A,n-2);

return ((A[0][0]\*4)%mod+(A[0][1]\*2)%mod+(A[0][2]\*1))%mod;

}

main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

if(n==1)

{cout<<"1\n";continue;}

else if(n==2)

{cout<<"2\n";continue;}

else if(n==3)

{cout<<"4\n";continue;}

cout<<ans(n)<<"\n";

}

}