```cpp
//*****************************************************************//
//*************string  Author = "SACHIN SAINI " *****************//
//*****************************************************************//

#include<bits/stdc++.h>
using namespace std;
long long find_fibonacci(long long);//function to find nth Fibonacci number
void calculatePowerofFn(long long [2][2],long long);//function to calculate power
void multiply(long long [2][2],long long[2][2]);//multiply function to multiply to matrices
#define mod 1000000007  //define mod with 1000000007 so we can use mod in place of 1000000007
int main()
{
    int t;
    cin>>t; //test case
    while(t--)
    {
        long long n;
        cin>>n;  //nth number is to be find
        cout<<find_fibonacci(n)<<endl;
    }
    return 0;
}
long long find_fibonacci(long long n)
{
    //We are using matrix {{1,1},{1,0}} to calculate the fibonacci number
    //We will calculate F[n] using {{1,1},{1,0}}^n  then after doing power answer will be
    F[0][0];
    long long F[2][2]={{1,1},{1,0}}; //fibonacci array of which we will calculate power
    if(n==0||n==1) //Base case
    {
        return 0LL; // ending with 'LL' makes it long long
    }
    else
    {
        calculatePowerofFn(F,n-1); //calling function calculate Power
    }
    return F[0][0]%mod; //returning the answer because answer is on the [0][0]th index
}
void calculatePowerofFn(long long F[2][2],long long n)
{
    if( n == 0 || n == 1)
        return;
    long long M[2][2] = {{1,1},{1,0}};

    calculatePowerofFn(F, n/2); //call recursively to find power fast
    multiply(F, F);

    if (n%2 != 0)
        multiply(F, M);
}
void multiply(long long F[2][2],long long M[2][2])
{
    //x,y,z,w are storing the value of matrix [0][0],[0][1],[1][0],[1][1] respectively;
    long long  x=((F[0][0]*M[0][0])%mod+(F[0][1]*M[1][0])%mod)%mod;
    long long  y=((F[0][0]*M[0][1])%mod+(F[0][1]*M[1][1])%mod)%mod;
    long long  z=((F[1][0]*M[0][0])%mod+(F[1][1]*M[1][0])%mod)%mod;
    long long  w=((F[1][0]*M[0][1])%mod+(F[1][1]*M[1][1])%mod)%mod;
    F[0][0]=x;
    F[0][1]=y;
    F[1][0]=z;
    F[1][1]=w;
}
```