**Ways To Tile A Floor**

Show Topic Tags   

[Amazon](http://practice.geeksforgeeks.org/company/Amazon/)

Given a floor of dimensions 2 x W and tiles of dimensions 2 x 1, write code to find the number of ways the floor can be tiled. A tile can either be placed horizontally i.e as a 1 x 2 tile or vertically i.e as 2 x 1 tile.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is W.

**Output:**

Print number of ways the floor can be tiled in a separate line.

**Constraints:**

1 ≤ T ≤ 50  
1 ≤ W ≤ 70

**Example:**

Input  
2  
5  
3

Output  
8  
3

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/ways-to-tile-a-floor/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication241;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Iterator;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication241 {

/\*\*

\* @param args the command line arguments

\*/

static long fib(int n)

{

/\* Declare an array to store Fibonacci numbers. \*/

long[] f = new long[(n+1)];

int i;

/\* 0th and 1st number of the series are 0 and 1\*/

f[0] = 0;

f[1] = 1;

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

{

/\* Add the previous 2 numbers in the series

and store it \*/

f[i] = f[i-1] + f[i-2];

}

return f[n];

}

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int w = Integer.parseInt(br.readLine());

System.out.println(fib(w+1));

}

}

}