

# 题目

## 题目描述

在一次愉快的网上摸鱼中， `hls` 突然发现了一个神奇的数列。这个数列  $H_n$  满足如下递推式：

$$H_n = \begin{cases} \sum_{i=1}^n H_{i-1} H_{n-i}, & n \geq 2 \\ 1, & n = 0, 1 \end{cases}$$

现在 `hls` 想知道第  $i$  项的值是多少，你能帮帮他嘛？

由于答案可能很大，因此输出需要对 `998244353` 进行取模。

## 输入

本题测试点包含多组数据。

第一行有 1 个整数，表示数据组数  $Q$ 。

接下来  $Q$  行，每行 1 个整数  $n$ ，表示查询  $H_n$  的大小。

数据保证  $1 \leq Q \leq 1000, 1 \leq n \leq 1000$ 。

## 输出

对于每组数据，输出  $Q$  行，每行为对应  $H_n$  取模后的值。

# 思路

本题考察卡特兰数，若用常规递归方法，时间复杂度约为 $O(n^2)$ ,在oj上会tle，所以不能采用直接递归的方法 本人的思路是用数组打表，把 $H_1-H_n$ 记录下来

# 代码实现

```
#include <iostream>
#include <vector>
using namespace std;

const int MOD = 998244353;
const int MAX = 1005;

int main()
{
    int N;
    cin >> N;

    vector<int> dp(MAX, 0);
```

```
dp[0] = 1;
dp[1] = 1;
for (int i = 2; i < MAX; i++)
{
    for (int j = 1; j <= i; j++)
    {
        dp[i] = (dp[i] + 1LL * dp[j - 1] * dp[i - j]) % MOD;
    }
}

for (int k = 0; k < N; k++)
{
    int n;
    cin >> n;
    cout << dp[n] << endl;
}

return 0;
}
```

## 要注意的细节

---

本题有大量的相乘累加，如果先算出结果再对 998244353 求模，很容易出现溢出。可以通过 1LL \* 或者在每个计算过程都 %Mod 来避免