

Dinner Games

Filename: dinner

John, Stephen, and Nick like to eat dinner together sometimes, but they've found one aspect of these dinners to be boring - paying the check! Splitting it evenly is simple: just divide by three. The most interesting part of this method is when the check doesn't split evenly and someone may have to pay a penny or two more than the others. Of course, this is still hardly interesting.

John has devised a game to make paying the check more fun! John will only pay with \$10 bills, Nick will only pay with \$5 bills, and Stephen will only pay with \$2 bills (assume everyone has an ample supply of their respective bill type to be capable of paying more than the entire bill). They will each keep laying down bills in any order in a pile until that pile adds up exactly to the total on the check (luckily, they only go to restaurants that are priced so this actually works!). They are all smart, so they never make mistakes that force the sum over the total.

Now Nick is worried: how fast will this game get boring? Or, more specifically, given an amount representing the check total, how many different piles can be made with this method to sum up to exactly this amount? A pile is defined as different from another if there is either a different number of one or more of the bills or if the bills occur in a different order. Bills of the same denomination are indistinguishable, though. So, for example, all of the following piles are different from each other for various reasons: $2+2+5$, $2+5+2$, $5+2+2$, $5+5+5$, $2+5$, $2+5+10$, $10+10$.

The Problem:

Given the amount of the check total, how many different ways are there for John, Stephen, and Nick to make piles that sum up to exactly that amount?

The Input:

The first line of input will be a positive number, c , the number of checks for which they want an answer. Following this will be c lines each describing one check. Each of these lines will contain an integer, t ($1 \leq t \leq 100$), representing the total amount of this check. There will be at least one way to pay each check using only the bills that John, Stephen, and Nick possess.

The Output:

For each check, output a line of the form "Dinner # i : s " where i is the number of the check in the input (starting with 1) and s is the number of different ways to sum up to exactly the check total using the method described above. *Note that this total could be quite large.*

Sample Input:

3
2
7
100

Sample Output:

Dinner #1: 1
Dinner #2: 2
Dinner #3: 9898980705