Problem 4: Look After The Pennies

Difficulty: Easy

Author: Ben Fenton, Ampthill, Reddings Wood, United Kingdom

Problem Background

A proverb from the United Kingdom says, "Look after the pennies, and the pounds will look after themselves." In other words, if you take care to consistently save money in small ways, you'll end up saving a much larger amount of money over the long term. Charities often take this advice and will partner with stores to ask customers to "round up" their bills to the nearest whole number and donate the extra change. Some banks also offer similar services with savings accounts.

Problem Description

You're working with a new banking company, Unlimited Finances, Ltd., to develop software to manage their account systems. Whenever customers use their credit or debit cards, the bank will round up the charge to the nearest whole number. The extra change will be placed in a designated savings account, where it can start earning interest. For example, if a customer pays a £3.54 bill with their card, the bank will charge the customer £4.00; the original £3.54 goes to whomever the customer was paying, and the extra £0.46 goes to the customer's savings account.

Given a list of charges from a customer, your program must generate an account statement listing the actual amount the customer was charged for each expense, followed by the total amount that was added to the customer's savings account.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A line containing a positive integer, X, representing the number of charges
- X lines, each containing a positive decimal value representing individual charges made by the customer

```
2
3
1.23
4.56
7.89
4
1.99
5.23
186.42
1234.56
```

Sample Output

For each test case, your program must print X lines, each containing an integer value representing the actual charges applied to the customer's account, in the order presented. Your program must then print one additional line containing a decimal value, representing the total amount transferred to the customer's savings account. Round this value to two decimal places and include any trailing zeros.

```
2
5
8
1.32
2
6
187
1235
1.80
```