2670 Euro Efficiency

On January 1st 2002, The Netherlands, and several other European countries abandoned their national currency in favour of the Euro. This changed the ease of paying, and not just internationally.

A student buying a 68 guilder book before January 1st could pay for the book with one 50 guilder banknote and two 10 guilder banknotes, receiving two guilders in change. In short: $50 + 10 + 10 - 1 - 1 = 68$. Other ways of paying were: $50 + 25 - 5 - 1 - 1$, or $100 - 25 - 5 - 1 - 1$. Either way, there are always 5 units (banknotes or coins) involved in the payment process, and it could not be done with less than 5 units.

Buying a 68 Euro book is easier these days: $50 + 20 - 2 = 68$, so only 3 units are involved. This is no coincidence; in many other cases paying with euros is more efficient than paying with guilders. On average the Euro is more efficient. This has nothing to do, of course, with the value of the Euro, but with the units chosen. The units for guilders used to be: 1, 2.5, 5, 10, 25, 50, whereas the units for the Euro are: 1, 2, 5, 10, 20, 50.

For this problem we restrict ourselves to amounts up to 100 cents. The Euro has coins with values 1, 2, 5, 10, 20, 50 eurocents. In paying an arbitrary amount in the range [1, 100] eurocents, on average 2.96 coins are involved, either as payment or as change. The Euro series is not optimal in this sense. With coins 1, 24, 34, 39, 46, 50 an amount of 68 cents can be paid using two coins. The average number of coins involved in paying an amount in the range [1, 100] is 2.52.

Calculations with the latter series are more complex, however. That is, mental calculations. These calculations could easily be programmed in any mobile phone, which nearly everybody carries around nowadays. Preparing for the future, a committee of the European Central Bank is studying the efficiency of series of coins, to find the most efficient series for amounts up to 100 eurocents. They need your help.

Write a program that, given a series of coins, calculates the average and maximum number of coins needed to pay any amount up to and including 100 cents. You may assume that both parties involved have sufficient numbers of any coin at their disposal.

Input

The first line of the input contains the number of test cases. Each test case is described by 6 different positive integers on a single line: the values of the coins, in ascending order. The first number is always 1. The last number is less than 100.

Output

For each test case the output is a single line containing first the average and then the maximum number of coins involved in paying an amount in the range [1, 100]. These values are separated by a space. As in the example, the average should always contain two digits behind the decimal point. The maximum is always an integer.

Sample Input

```
3
1 2 5 10 20 50
```

```
1 24 34 39 46 50
1 2 3 7 19 72
```

Sample Output

```
2.96 5
2.52 3
2.80 4
```