

## Problem X : Arithmetic Progression

Input from file : `x.in`

Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)

Execution time limit: 2 seconds

Andy Mabini would like to automatically determine the  $n$ th element of an arithmetic progression  $a_1, a_2, a_3, \dots, a_n$  and the sum of the elements  $a_1, a_2, a_3, \dots, a_n$ . The difference of  $a_i$  and  $a_{i+1}$  is constant  $d$  for all  $i$  where  $1 \leq i < n$ .

For example, Andy was given the arithmetic progression  $2, 5, 8, \dots$  and was asked to determine the 5th element, i.e.,  $a_5$ . He observed that the difference of 2 and 5 is the same as the difference of 5 and 8, which is 3. Hence the 4th element must be  $8 + 3 = 11$  and the 5th element must be  $11 + 3 = 14$ . The sum of  $3 + 5 + 8 + 11 + 14$  is 40.

In this example, we have  $n = 5$  elements, where  $a_1 = 2$ ,  $a_2 = 5$ ,  $a_3 = 8$ ,  $a_4 = 11$ , and  $a_n = a_5 = 14$ .

### INPUT FORMAT

The input starts with an integer  $N$  followed by  $N$  input cases where  $0 < N < 101$ . Each input case is composed of  $a_1$ ,  $a_2$ ,  $a_3$ , and  $n$ . The range of values of  $a_1$  is -1,000 and 1,000. The range of values of  $n$  is 1 to 1,000.

### OUTPUT FORMAT

For every input case, output  $a_n$  and the sum  $a_1 + a_2 + \dots + a_n$ .

### SAMPLE INPUT

```
3
2 5 8 5
-1.5 -0.5 0.5 10
1 0 -1 5
```

### SAMPLE OUTPUT

```
14 40
7.5 30
-3 -5
```

## Problem Y : Counting Primes

Input from file : `y.in`

Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)

Execution time limit: 2 seconds

Given two positive integers  $a$  and  $b$  where  $0 < a < b$ , determine the number of prime numbers from  $a$  to  $b$ .

### INPUT FORMAT

The input starts with an integer  $N$  followed by  $N$  input cases where  $0 < N < 101$ . Each input case is composed of two integers  $a$  and  $b$  where  $0 < a < b \leq 1,000,000$ .

### OUTPUT FORMAT

For every input case, output the number of primes from  $a$  to  $b$ .

### SAMPLE INPUT

```
3
2 3
2 5
3 10
```

### SAMPLE OUTPUT

```
2
3
3
```