# B: **Balance Game**

Source file name: `balance.c, balance.cpp`, `balance.java`, *or* `balance.py`
Author: Rodrigo Cardoso

Consider a puzzle with three types of tokens. Tokens of the same type have the same weight and any two tokens of different type have different weight. There are M copies of each token type. There is a two-plate scale where tokens can be placed. There is also a given object with known weight W that can be put on the plates of the scale. The goal of the puzzle is to balance the scale with some of the tokens to check the weight of the object. If used, tokens of the same type must be allocated on the same plate.

For example, suppose that there are three token types with weights 20, 50, and 40 grams, respectively. Also, there are $M = 3$ copies of each token type. If the given object weights $W = 140$ grams, then its weight can be checked in five different ways (neglecting the order of the plates):
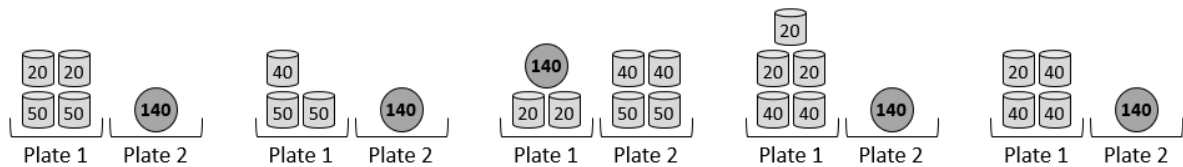
Five different ways to achieve the goal

However, it is impossible to measure the weight $W = 105$ with this set of tokens.

Your job is to write a computer program to determine the number of different ways to check the weight of the object.

Input

The input consists of several test cases. The first line of each test case contains two blank-separated integers M and W ($1 \le M \le 5\,000$ and $1 \le W \le 5\,000\,000$) indicating, respectively, the number of copies of each token type and the weight of the object to be checked. Then there is a line containing three distinct blank-separated integers n_1, n_2, and n_3 representing the weigths of each token type ($1 \le n_i \le 1\,000$, for each $1 \le i \le 3$, $n_1 \ne n_2$, $n_1 \ne n_3$, $n_2 \ne n_3$).

The input must be read from standard input.

Output

For each test case, print a single line indicating the number of different ways the given weight can be checked with the set of tokens.

The output must be written to standard output.

Sample Input	Sample Output
3 140	5
20 50 40	0
3 105	
20 50 40	