After long years of sharpening your programming skills and honing your physique you finally made it: as a member of the ProProgrammers team you've entered [Fort Boyard](https://en.wikipedia.org/wiki/Fort_Boyard_(TV_series))!

It's your turn to show who's the boss: you're the one who will be participating in the next challenge. Here are the rules of this challenge.

There's a treadmill hanging from the ceiling and going up. At its very top there are three buckets containing water. You can jog up the treadmill and perform one of the following operations:

* empty one of the buckets entirely and pour out all water from it;
* pour water from one bucket into another until either the first one is empty or the second one is full.

The buckets are connected to one arm of the scales, and the key that your team should reach is hanging from another. The key will be reachable only when the total amount of water in all three buckets is equal to the given volume.

The capacities (volumes) of all the buckets are stored in an array cap. Initially all three buckets are full, i.e. for each valid i the ith bucket has cap[i] units of water. You want to make sure that your efforts won't be vain and calculate in advance if it is possible to get the key. Given the cap array and the volume, determine if it is possible to obtain the volume amount of water in all three buckets performing only the allowed operations.

Example

* For cap = [1, 1, 1] and volume = 2, the output should be  
  runningWater(cap, volume) = true.

Initially there are 3 units of water in all the buckets. The only action you can perform is to empty the water from one of the buckets, which will leave you with 2 units. At this point, pouring water from one bucket to another won't produce a different total amount of water, so the only remaining option is emptying one of the remaining buckets, which will leave you with 1 final unit of water. Now the total amount of the poured water is equal to the given volume, so the answer is true.

* For cap = [16, 5, 3] and volume = 20, the output should be  
  runningWater(cap, volume) = false.

Here is the list of all possible amounts of water across all three buckets: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 24. As you can see, you can't obtain the given volume.

Input/Output

* **[execution time limit] 4 seconds (py3)**
* **[input] array.integer cap**

Individual capacity of each bucket (their volume), listed as three positive integers.

*Guaranteed constraints:*  
cap.length = 3,  
1 ≤ cap[i] ≤ 100.

* **[input] integer volume**

The required volume.

*Guaranteed constraints:*  
1 ≤ volume ≤ sum(cap[i]).

* **[output] boolean**

true if the given volume can be obtained using the allowed operations and false otherwise.