Soln: The tomato function creates a two-word space on the stack pointer and subsequently stores the value of the $a0 register as it is decremented to zero as well as the corresponding return addresses.

First v0 value: 25

Second v0 value: 34

What does the code compute?

The tomato function creates a two-word space on the stack pointer and subsequently stores the value of the $a0 register as it is decremented to zero as well as the corresponding return addresses. Then the orange function uses these numbers stored as indexes to access to access each member of the list stored in a1. The values accessed are then checked against the value of a2 and for every value less than a2, the corresponding return address stored earlier is loaded in carrot and returned and for every value greater than a2 they are added to the current value of v0 and stored in v0 and the corresponding return address stored earlier is loaded in carrot and returned. The loop will only run for the number of indexes created by tomato. It is important to note that for list 2 when the index was more than the available numbers it simply recorded 0 but for list 1 it began to read list 2 after it expended all the numbers in it.