The output of the code is

n + 1 + 4

n + 1 + 4

n + 1 + m

n + 1 + `m'

`n' + 1 + ``m''

The sum() macro takes in two arguments. The first argument is an argument to an incer() macro, while the second argument does not belong to or is not used by another macro.

Effectively, by calling the sum() macro, we are actually doing this: sum(`incer(`$1 + 1’) + $2’). When looking at the sum() macro call in this way, we can see that the first argument is surrounded by two sets of quotation marks while the second argument is only surrounded by one quotation mark.

When scanning, the machine removes one set of quotation marks per macro call for each of the arguments during the interpretation and output. When it sees the next quotation set after removing the quotation marks from the macro calls, it takes the argument literally. Since we are effectively calling two macros for the first argument and only one macro call for the second, the machine will remove two sets of quotation marks for the first argument and one set for the second argument. This is why by the third time we call the sum() where we are passing ``n’’ and ``m’’, the output is n + 1 + m instead of n + 1 + 4. The m is wrapped in two sets of quotation marks but since the second argument is only wrapped in one set in the definition of the macro, only one set is removed. The machine sees the next remaining quotation after removing the first set and then takes ‘m’ literally.