HOW TO ADD LARGE NUMBERS USING STACKS!

Rynz A. Daval
CompSci 22-A

Given two large numbers:

Place them into variables
operand_1 and ***operand_2.***

Let's say:

operand_1 *= 9367*
operand_2 *= 891*

Create:

A variable ***sum*** to store the sum.

Three stacks called ***stack_1*** and ***stack_2*** and ***stack_3***.

stack_1 is for ***operand_1.***

stack_2 is for ***operand_2.***

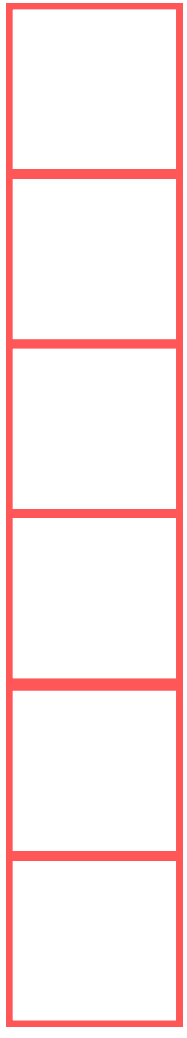
stack_3 is for **the sum.**

Create also a variable ***carry*** that contains a temporary value.

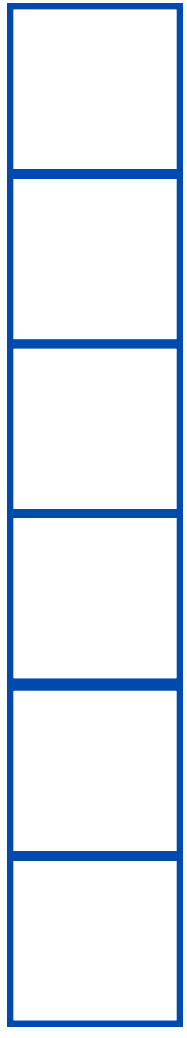
Declare all integer variables as 0.

Before Solution:

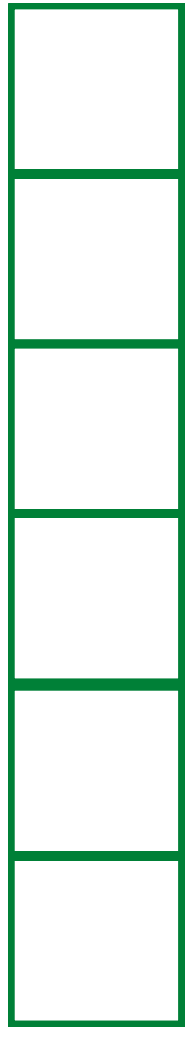
stack_1



stack_2



stack_3



variables:

operand_1

= 9367

operand_2

= 891

carry

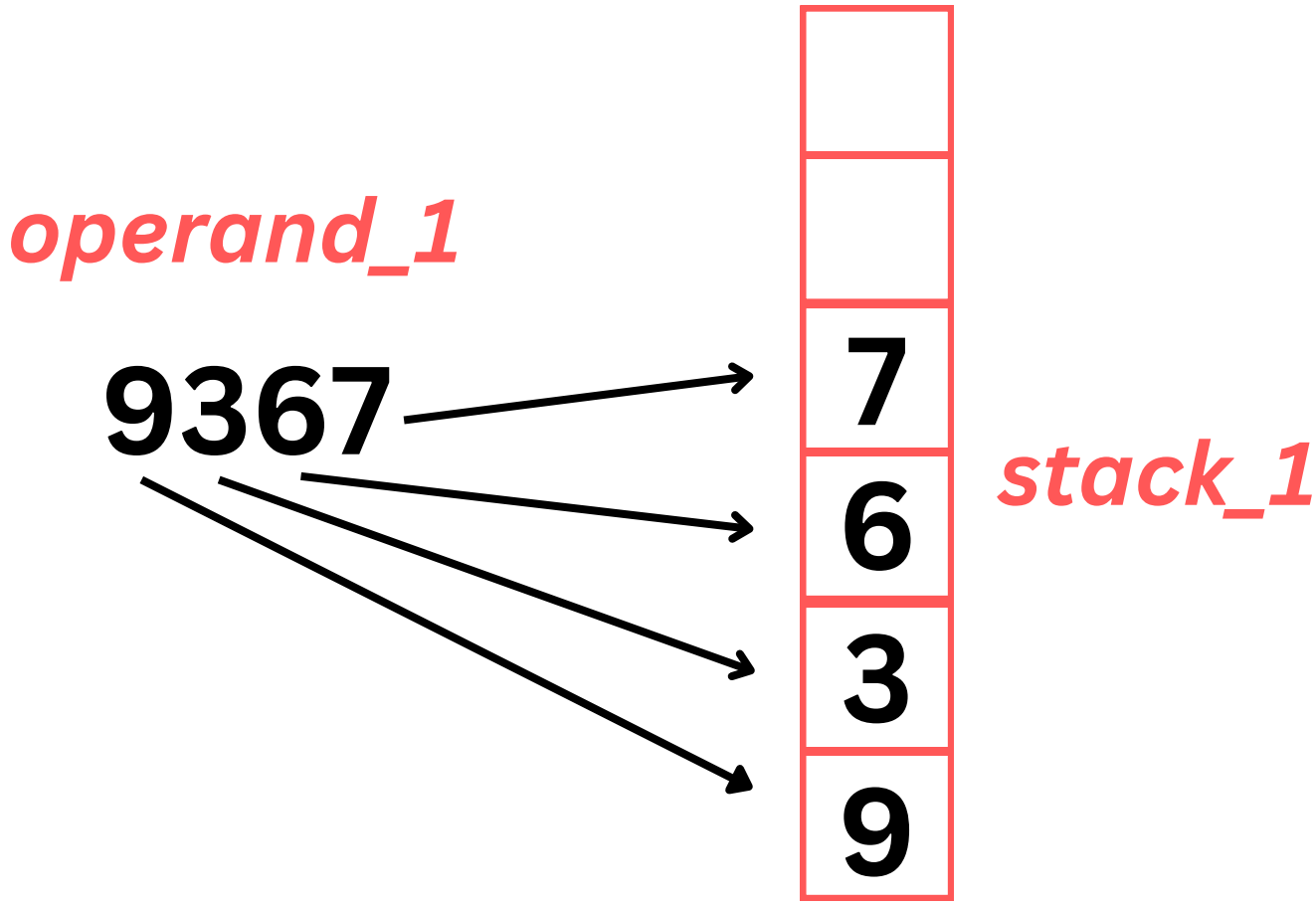
= 0

sum

= 0

Step 1:

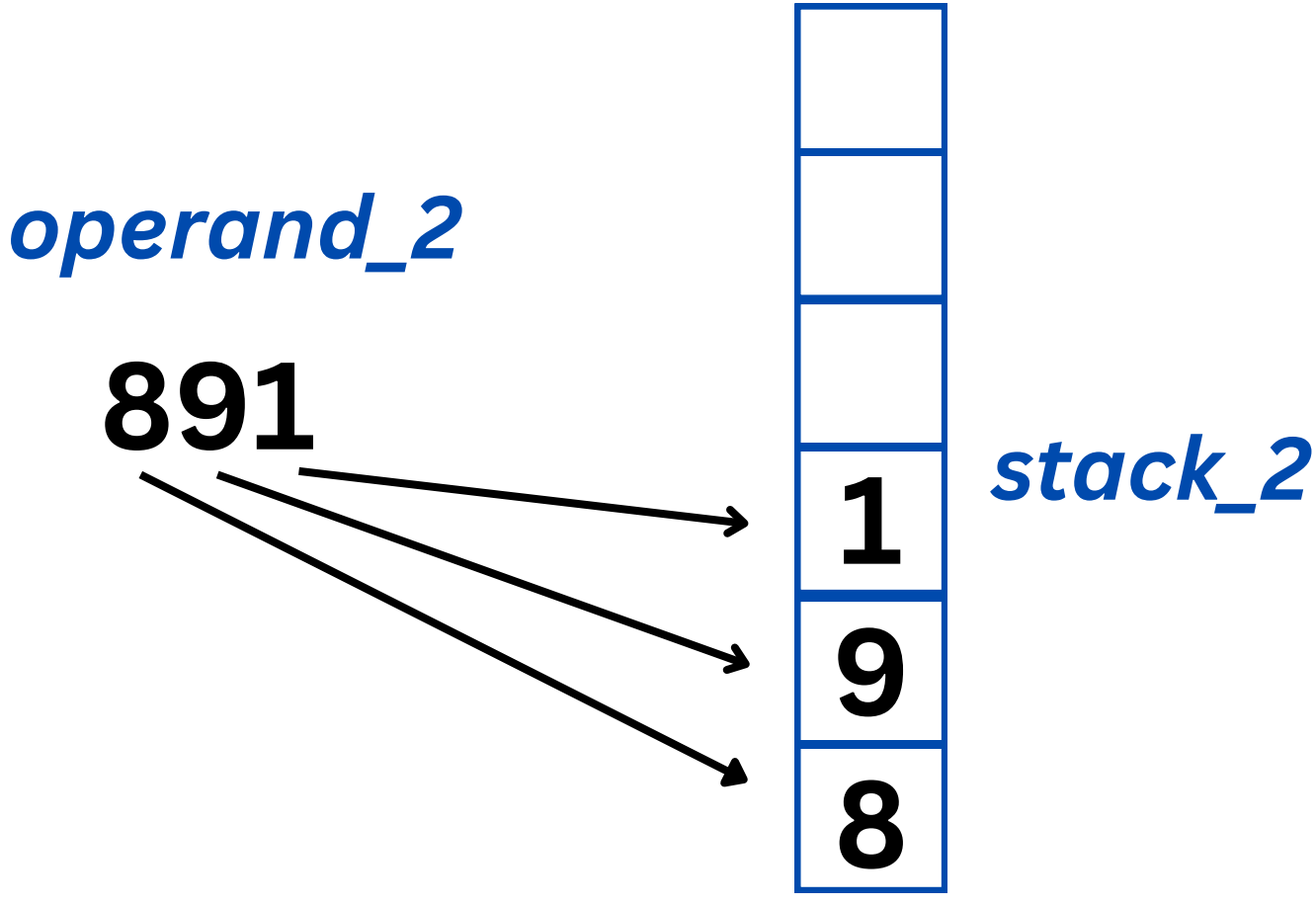
Push the digits from *operand_1* into *stack_1* from left to right, one by one.



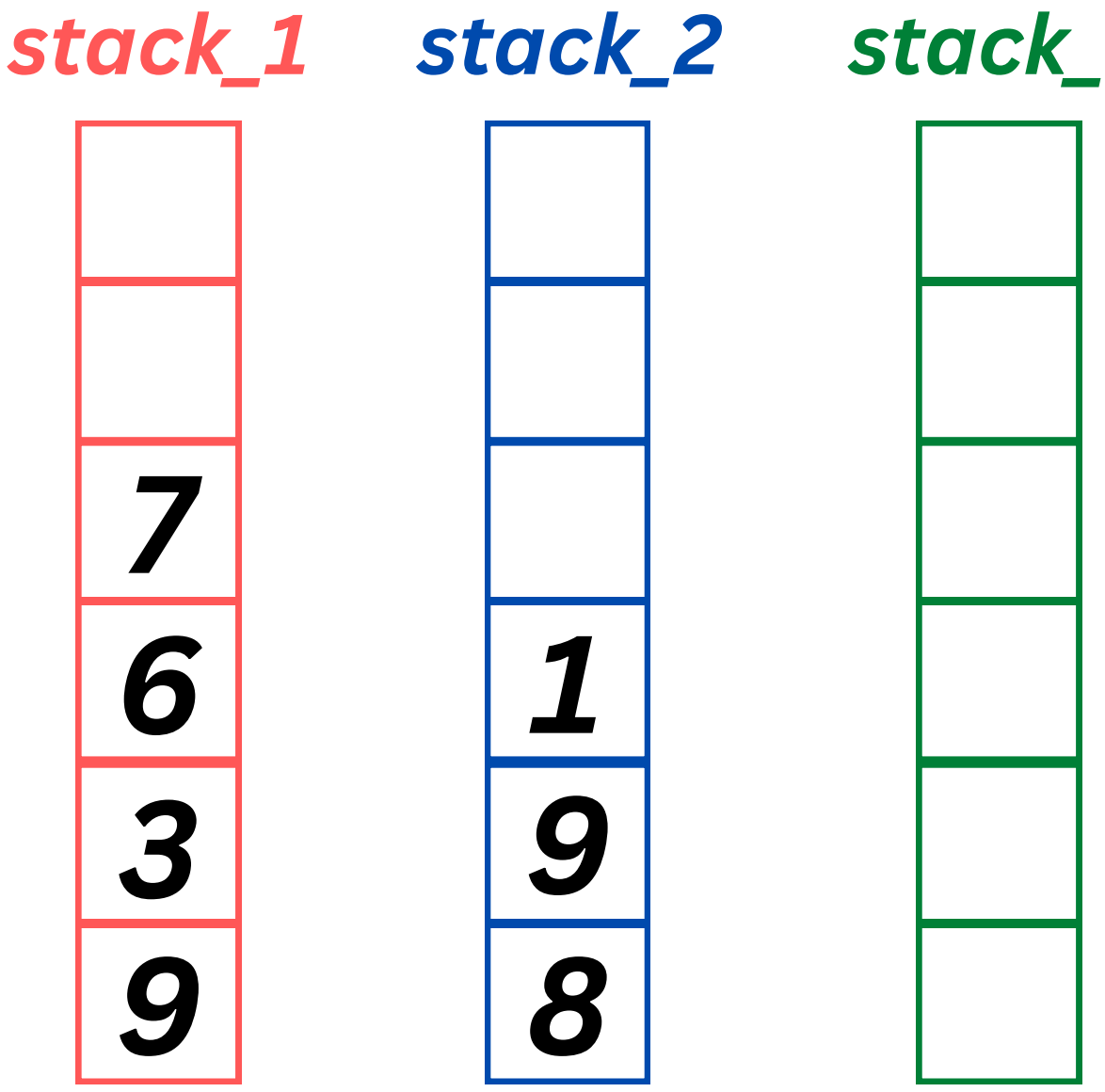
Divide operand_1 by 10 until it is no longer >10 to find its leftmost digit.

Step 2:

Repeat Step 1, this time for ***operand_2*** and ***stack_2.***



After Step 1 and 2:



variables:

operand_1

= 0

operand_2

= 0

carry

= 0

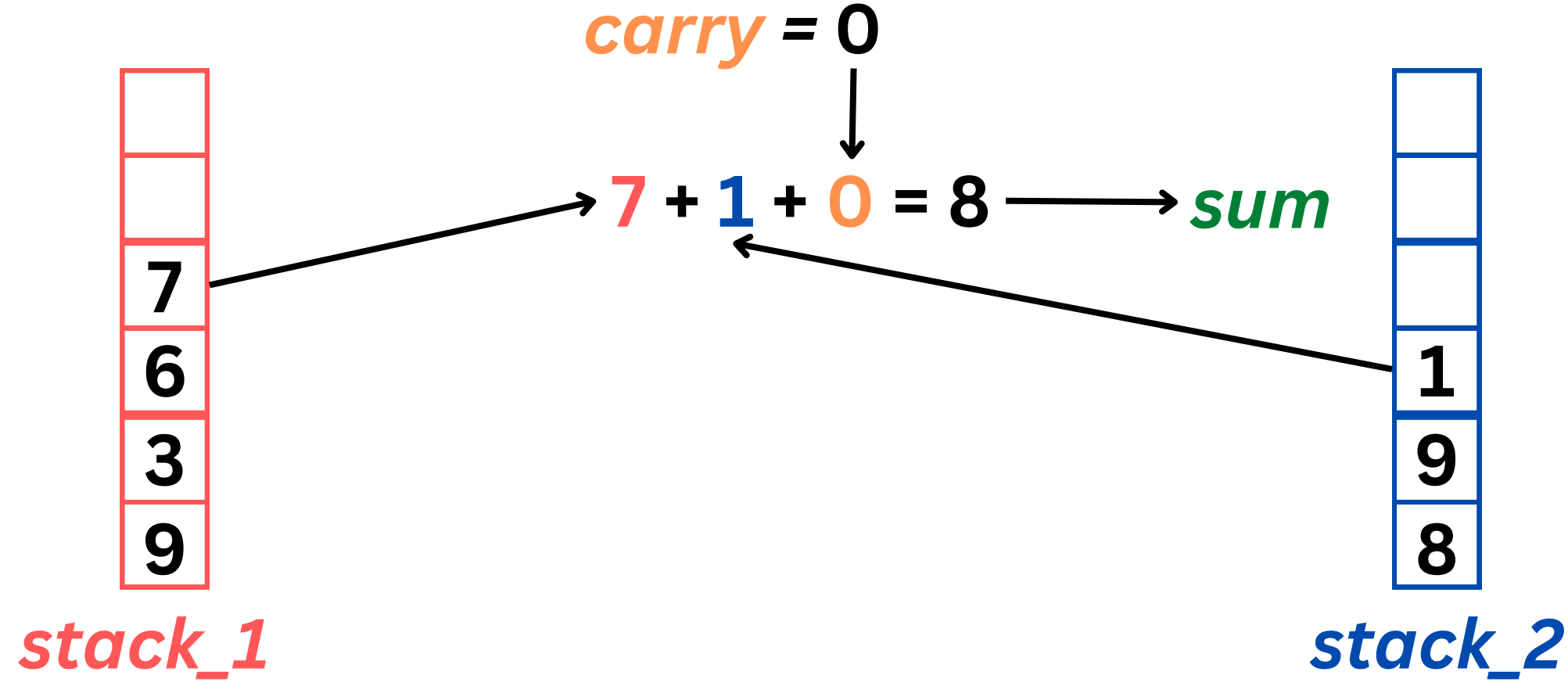
sum

= 0

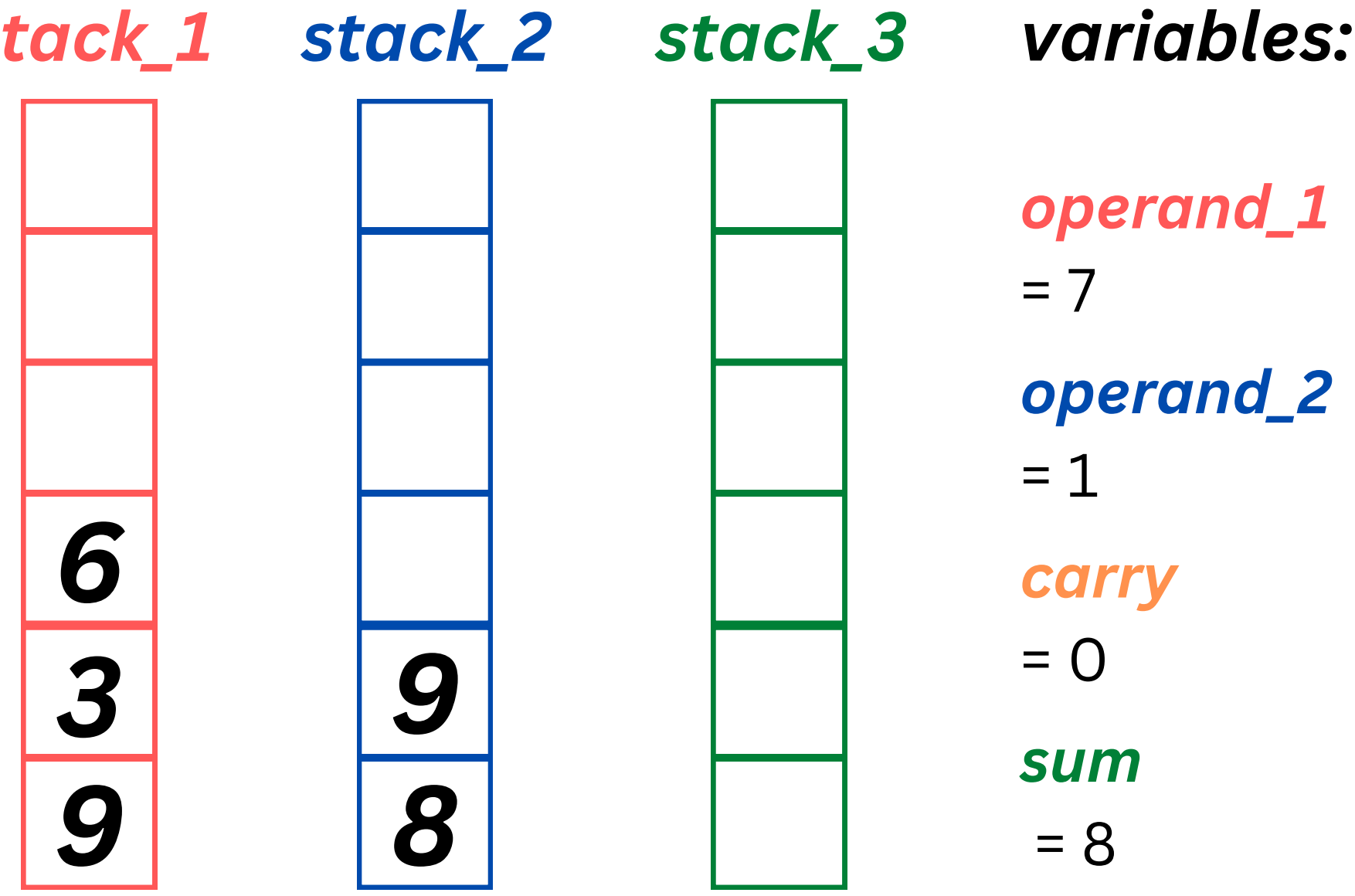
operand_1 and operand_2 becomes 0 after Step 1 and 2.

Step 3:

Pop a digit from the top of *stack_1* and *stack_2*. Then, add them together with variable *carry* and store result into *sum.*



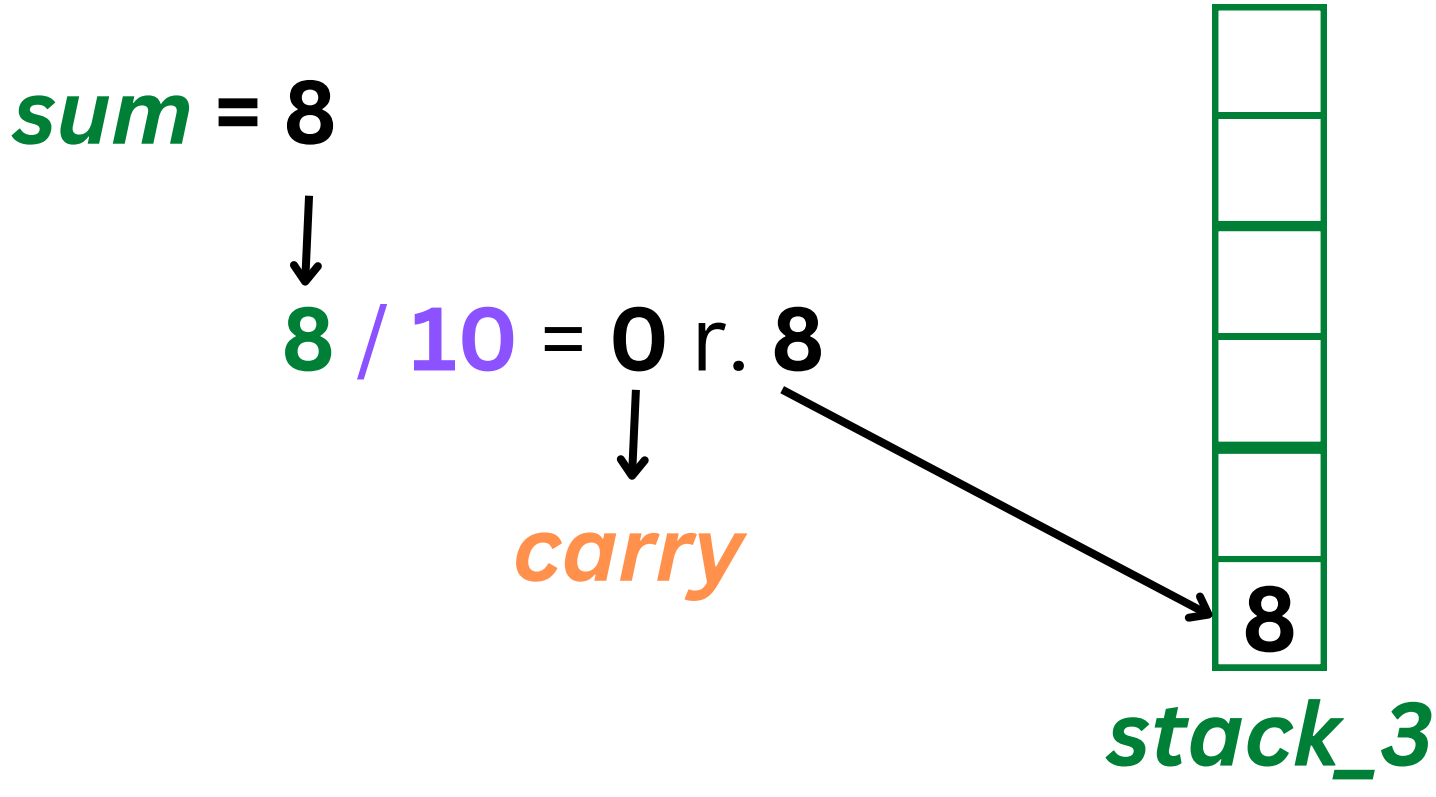
After Step 3:



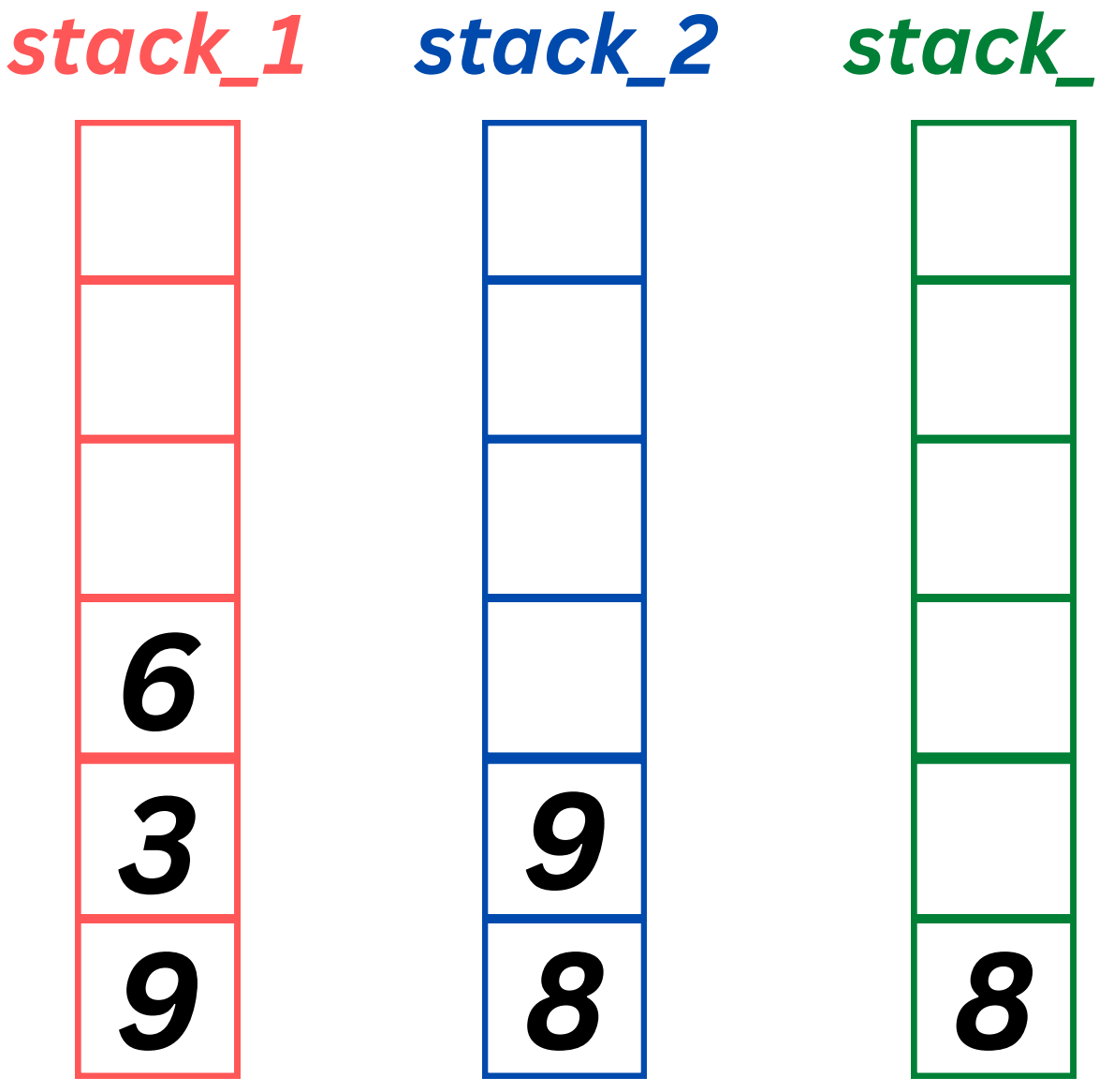
operand_1 and operand_2 can be used to temporarily store the popped values from stack_1 and stack_2.

Step 4:

Divide ***sum*** by 10 and store the quotient into ***carry***. The remainder of ***sum*** divided by 10 will be pushed into ***stack_3***.



After Step 4:



variables:

operand_1
= 7

operand_2
= 1

carry
= 0

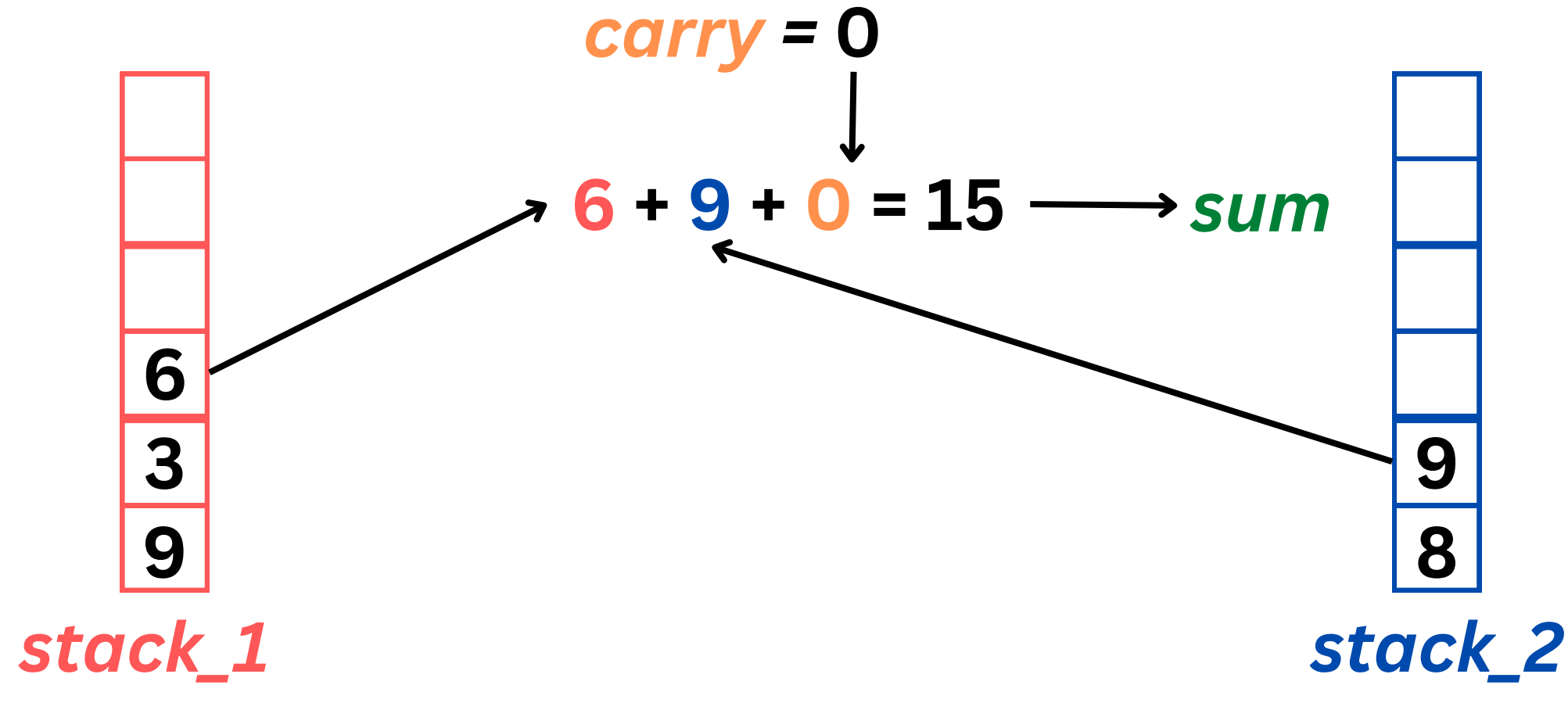
sum
= 8

Step 5:

Repeat Step 3 and 4 until ***stack_1*** and ***stack_2*** are empty and ***carry*** is equal to 0.

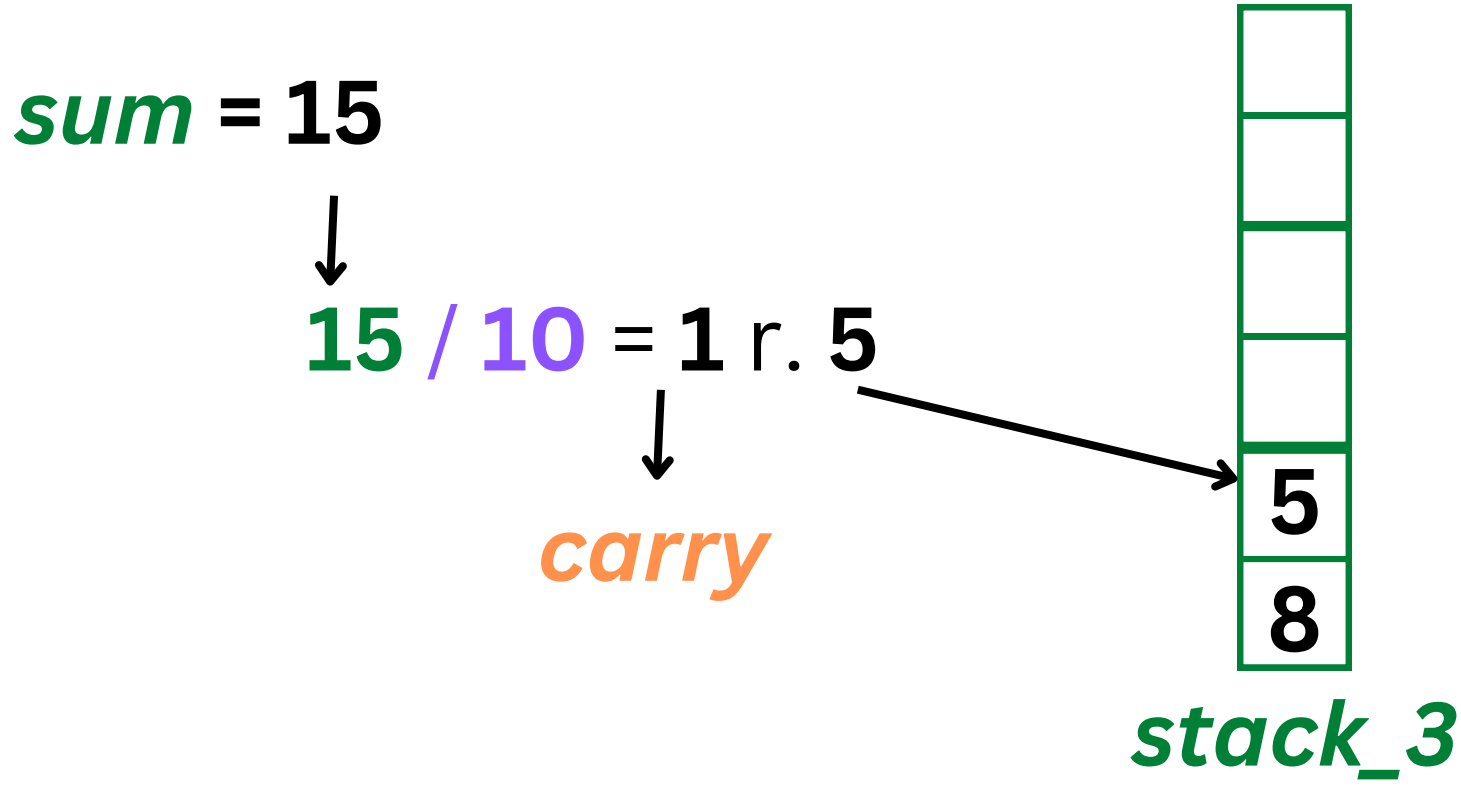
Step 5 iteration 1:

Pop a digit from the top of *stack_1* and *stack_2*. Then, add them together with variable *carry* and store result into *sum.*

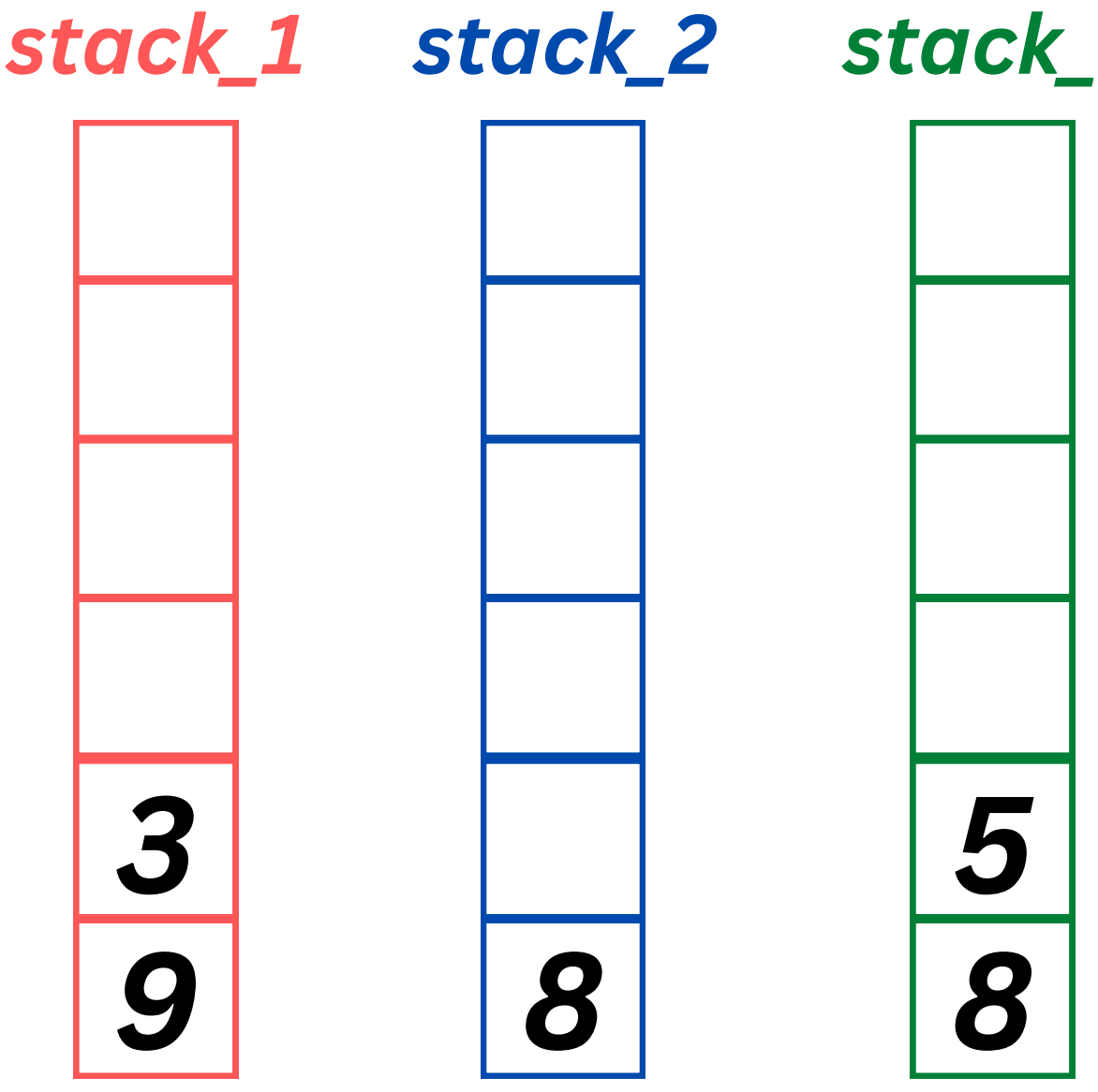


Step 5 iteration 1:

Divide ***sum*** by 10 and store the quotient into ***carry***. The remainder of ***sum*** divided by 10 will be pushed into ***stack_3***.



After Step 5 iteration 1:



variables:

operand_1
= 6

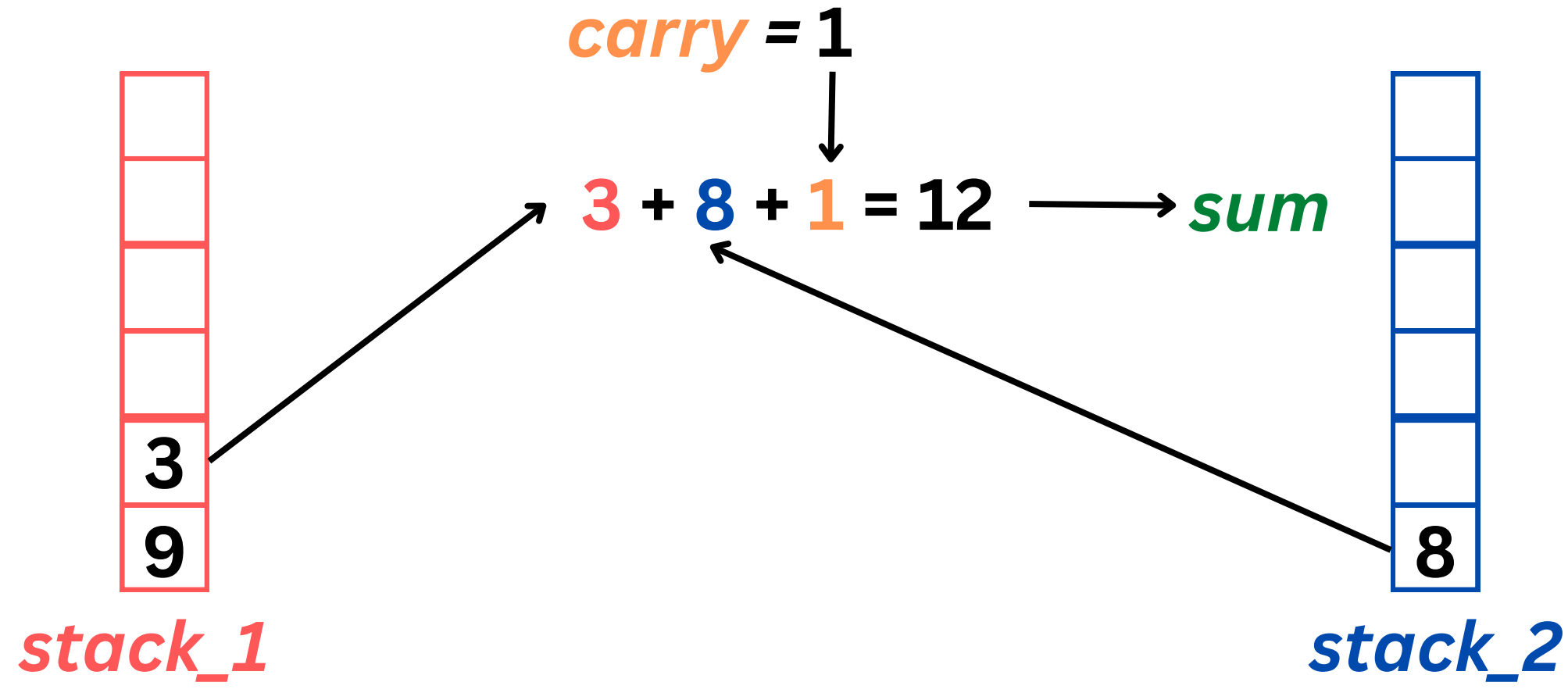
operand_2
= 9

carry
= 1

sum
= 15

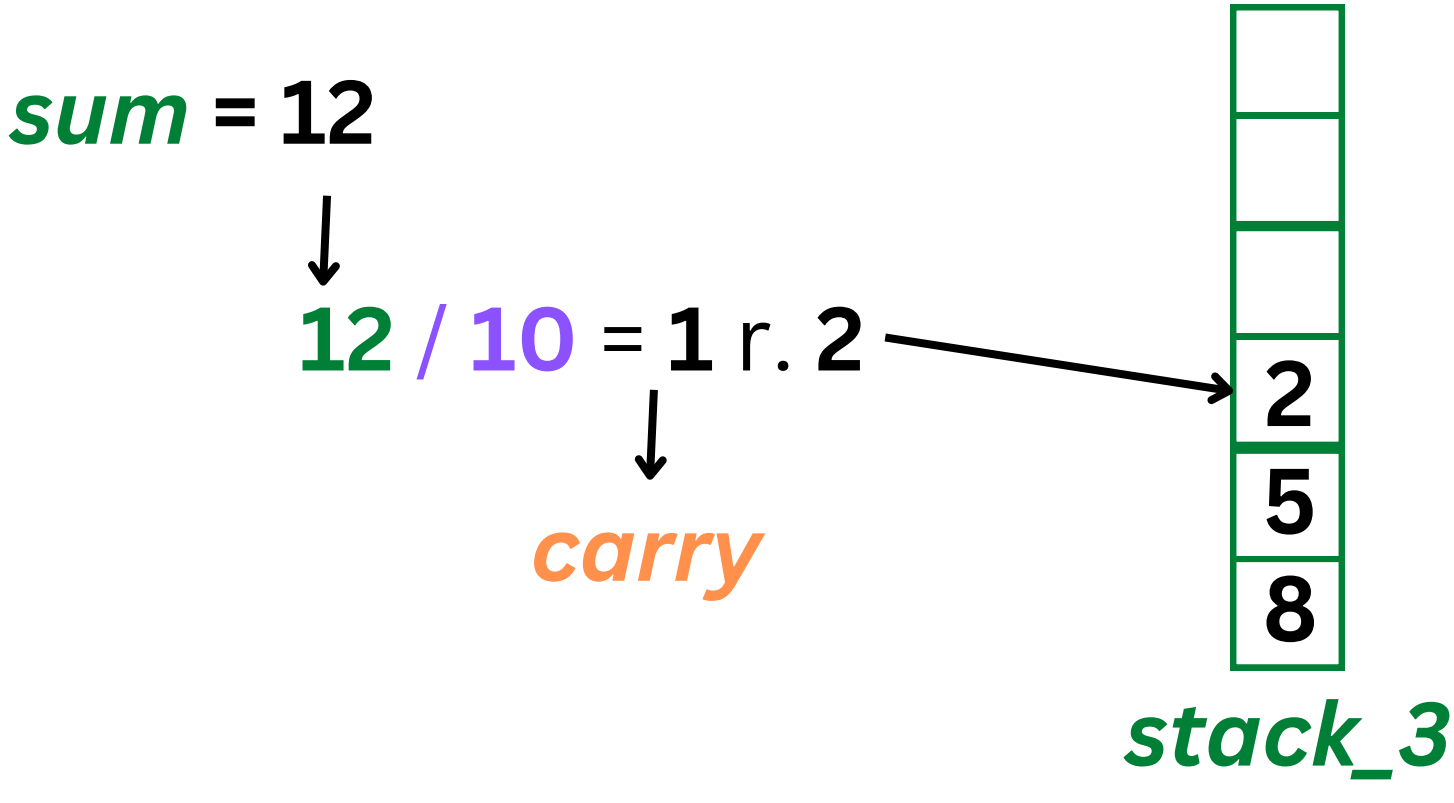
Step 5 iteration 2:

Pop a digit from the top of *stack_1* and *stack_2*. Then, add them together with variable *carry* and store result into *sum.*

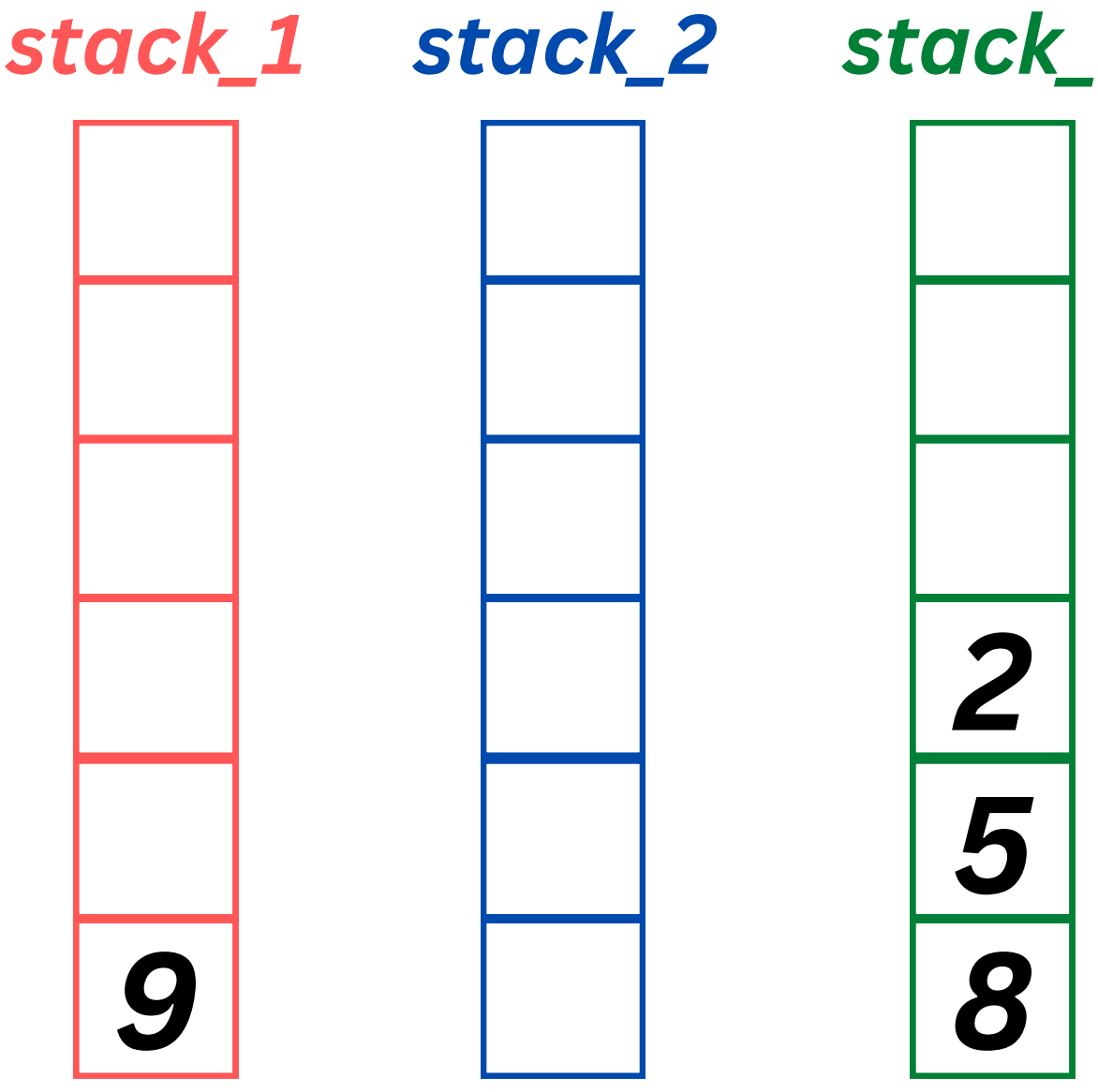


Step 5 iteration 2:

Divide ***sum*** by 10 and store the quotient into ***carry***. The remainder of ***sum*** divided by 10 will be pushed into ***stack_3***.



After Step 5 iteration 2:



variables:

operand_1

= 3

operand_2

= 8

carry

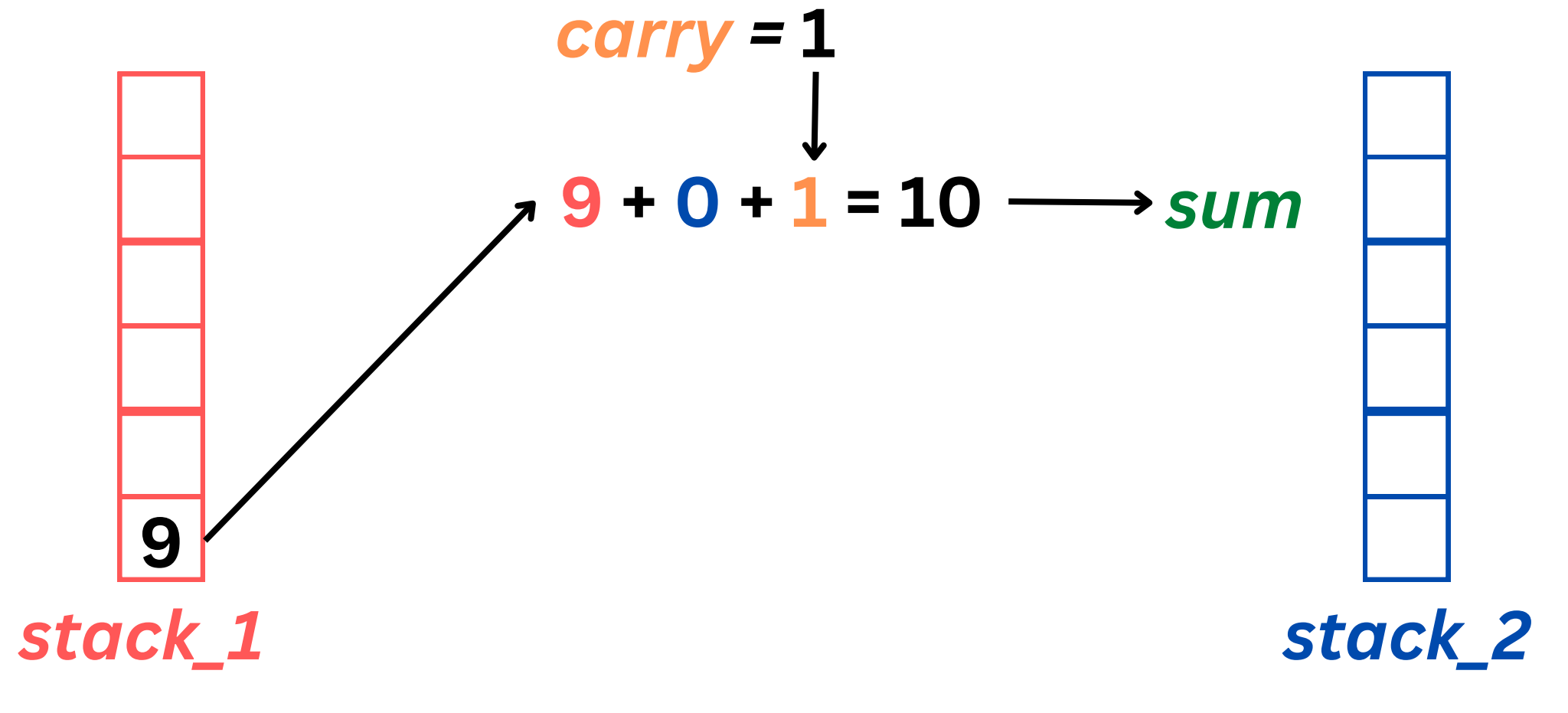
= 1

sum

= 12

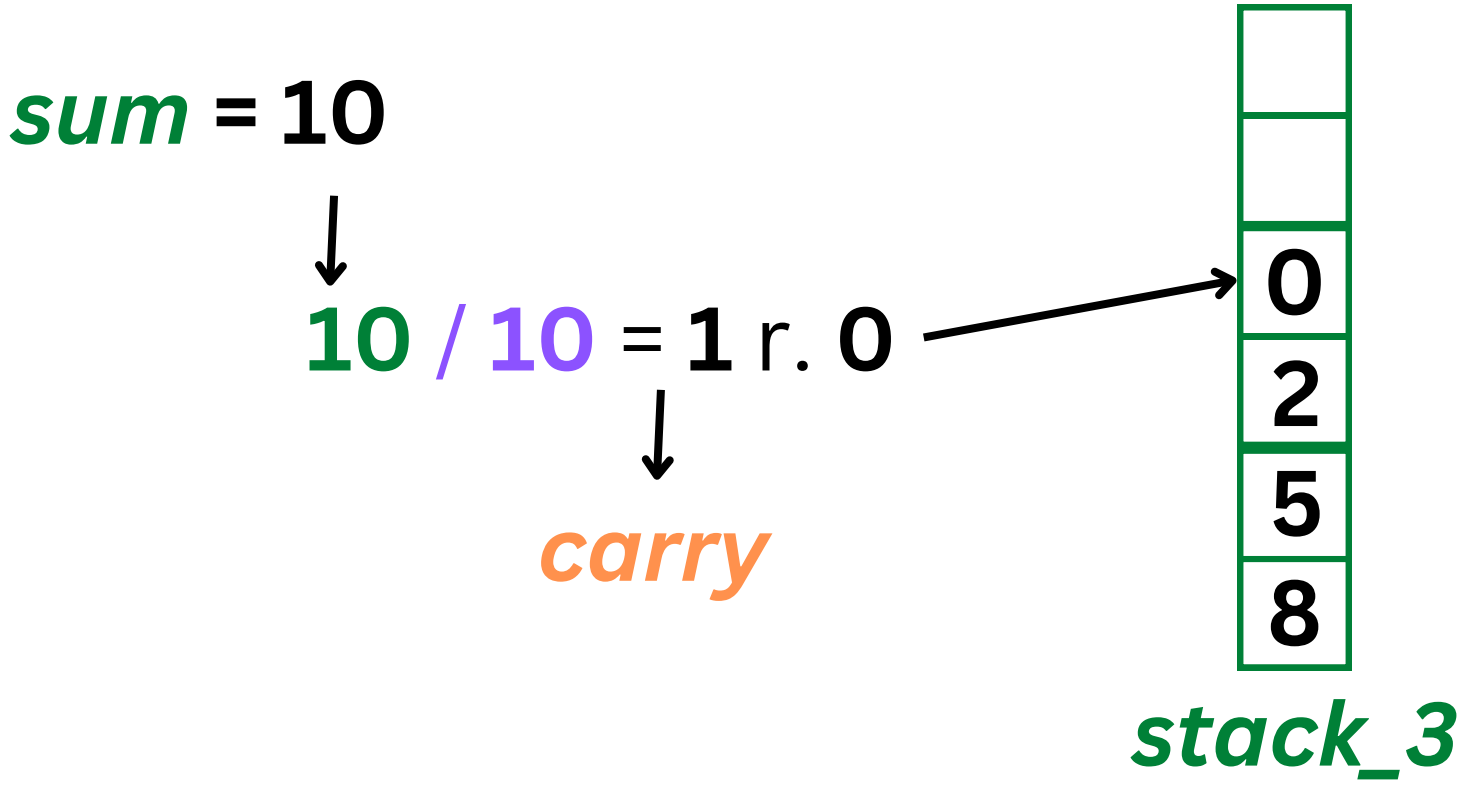
Step 5 iteration 3:

Pop a digit from the top of *stack_1* and *stack_2*. Then, add them together with variable *carry* and store result into *sum.*

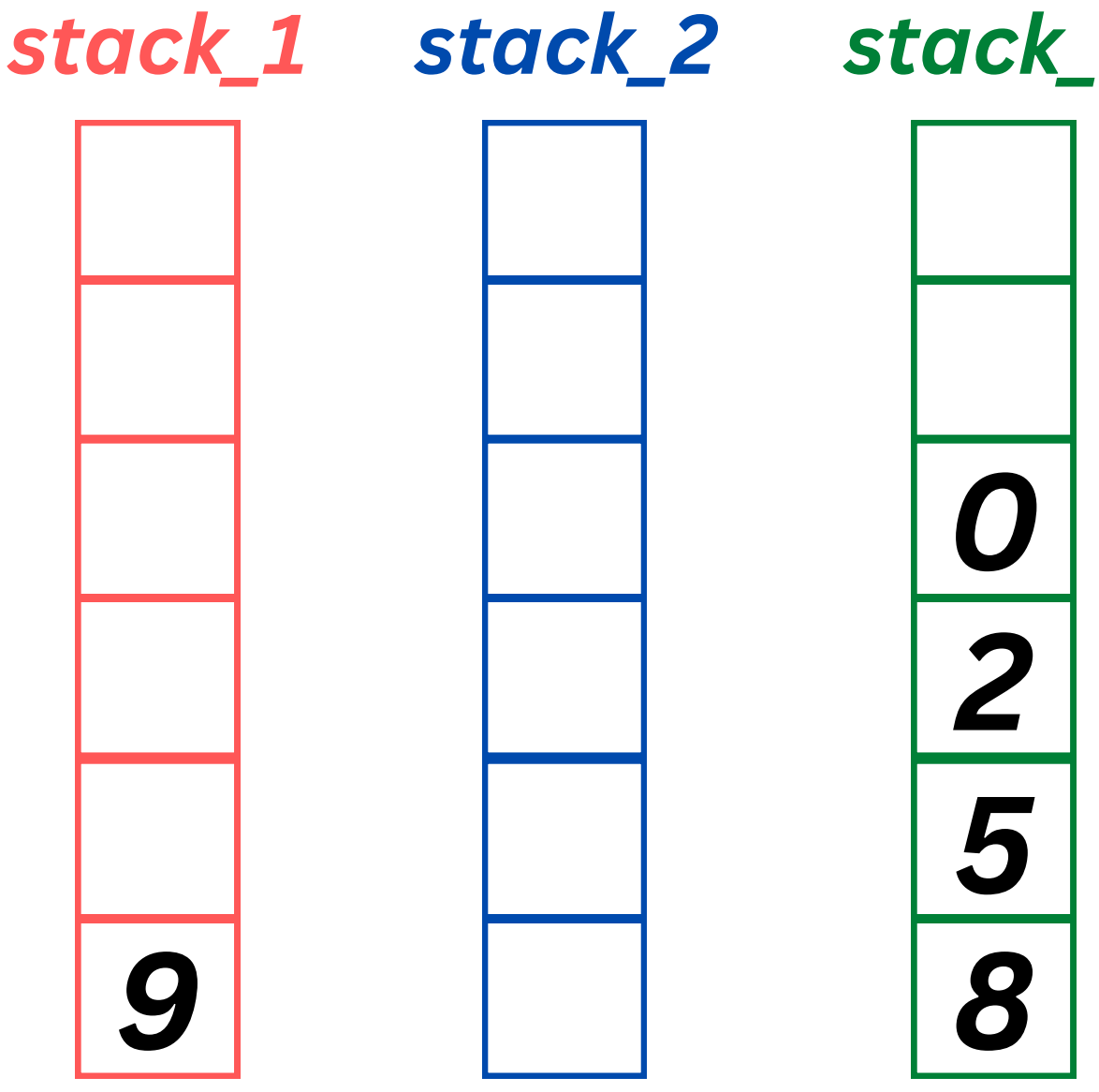


Step 5 iteration 3:

Divide ***sum*** by 10 and store the quotient into ***carry***. The remainder of ***sum*** divided by 10 will be pushed into ***stack_3***.



After Step 5 iteration 3:



variables:

operand_1
= 9

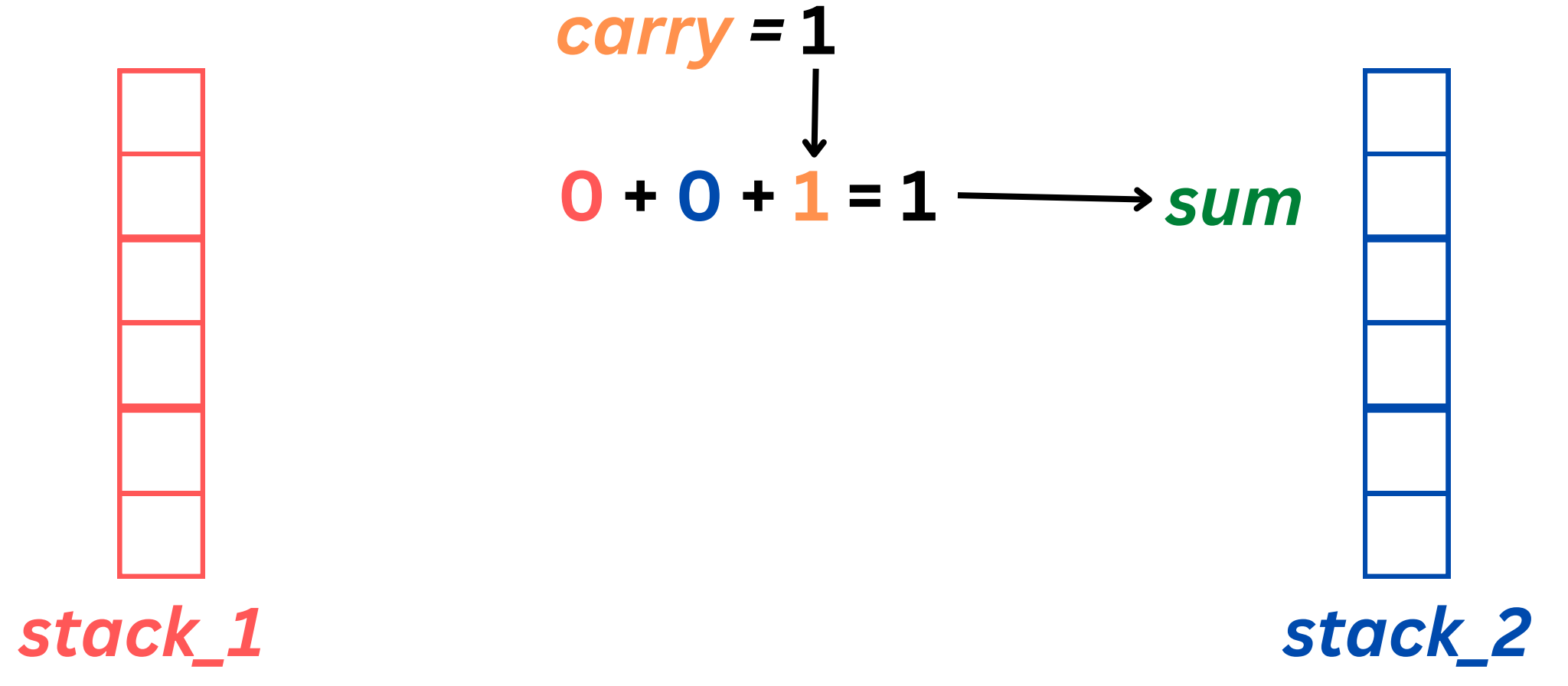
operand_2
= 0

carry
= 1

sum
= 10

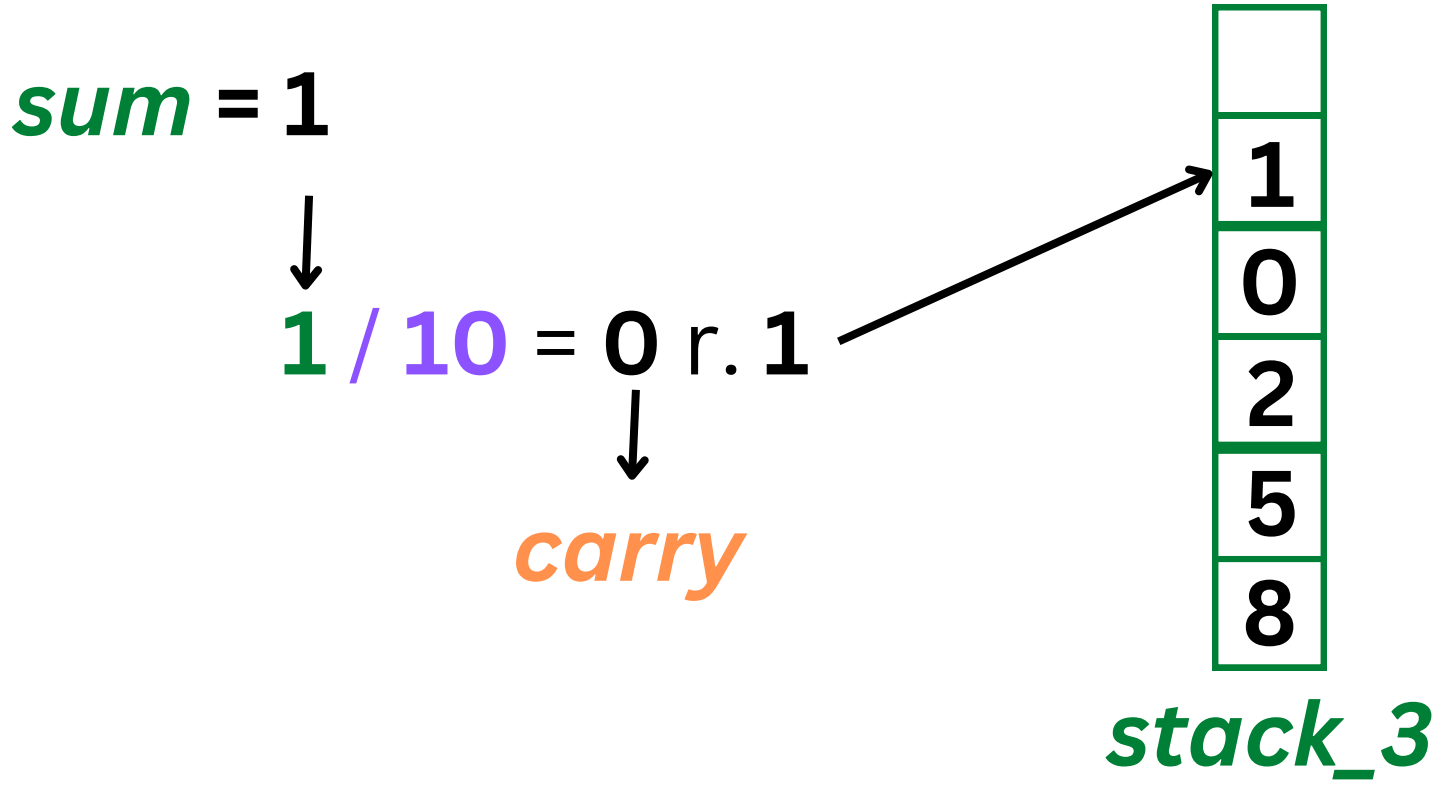
Step 5 iteration 4:

Pop a digit from the top of ***stack_1*** and ***stack_2***. Then, add them together with variable ***carry*** and store result into ***sum.***

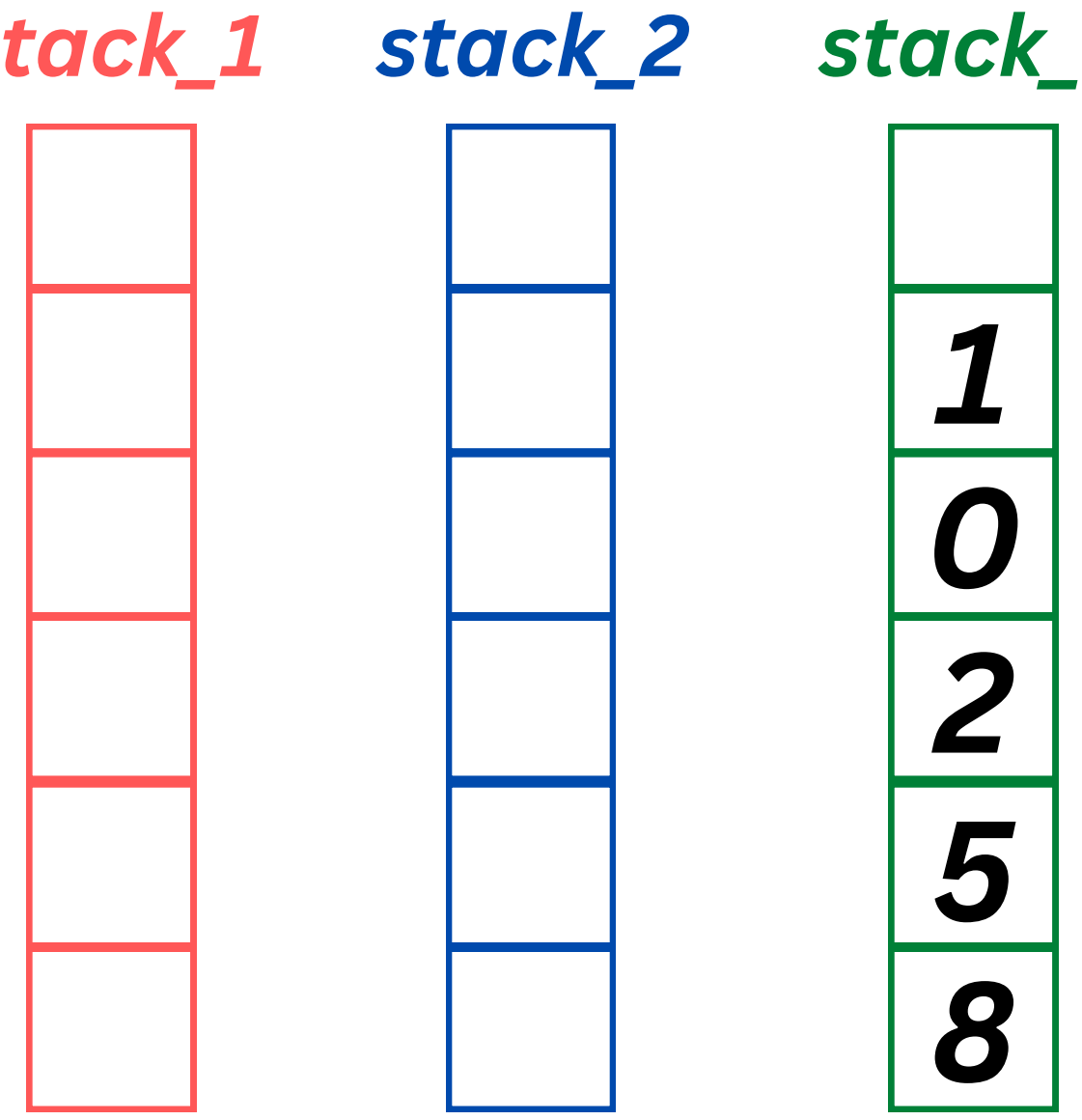


Step 5 iteration 4:

Divide ***sum*** by 10 and store the quotient into ***carry***. The remainder of ***sum*** divided by 10 will be pushed into ***stack_3***.



After Step 5 iteration 4:



variables:

operand_1
= 0

operand_2
= 0

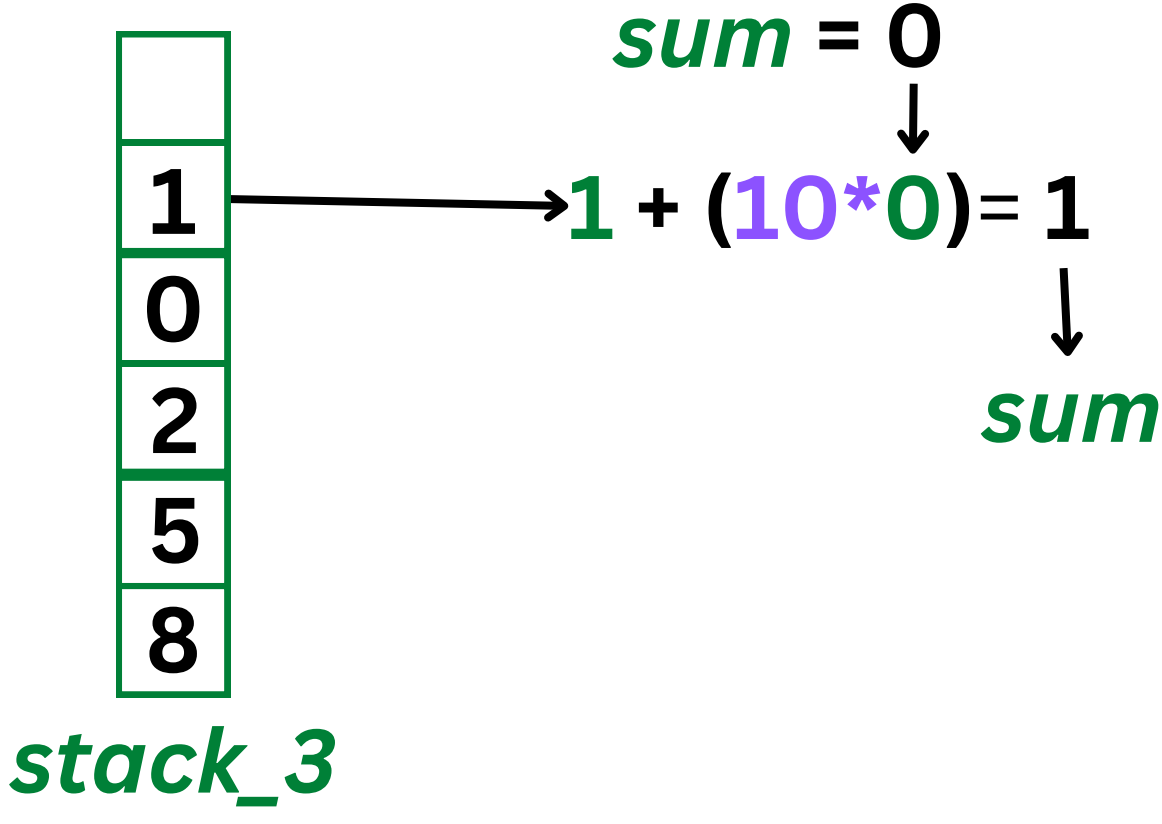
carry
= 0

sum
= 1

After iteration 4, stack_1 and stack_2 are now empty and carry is 0.

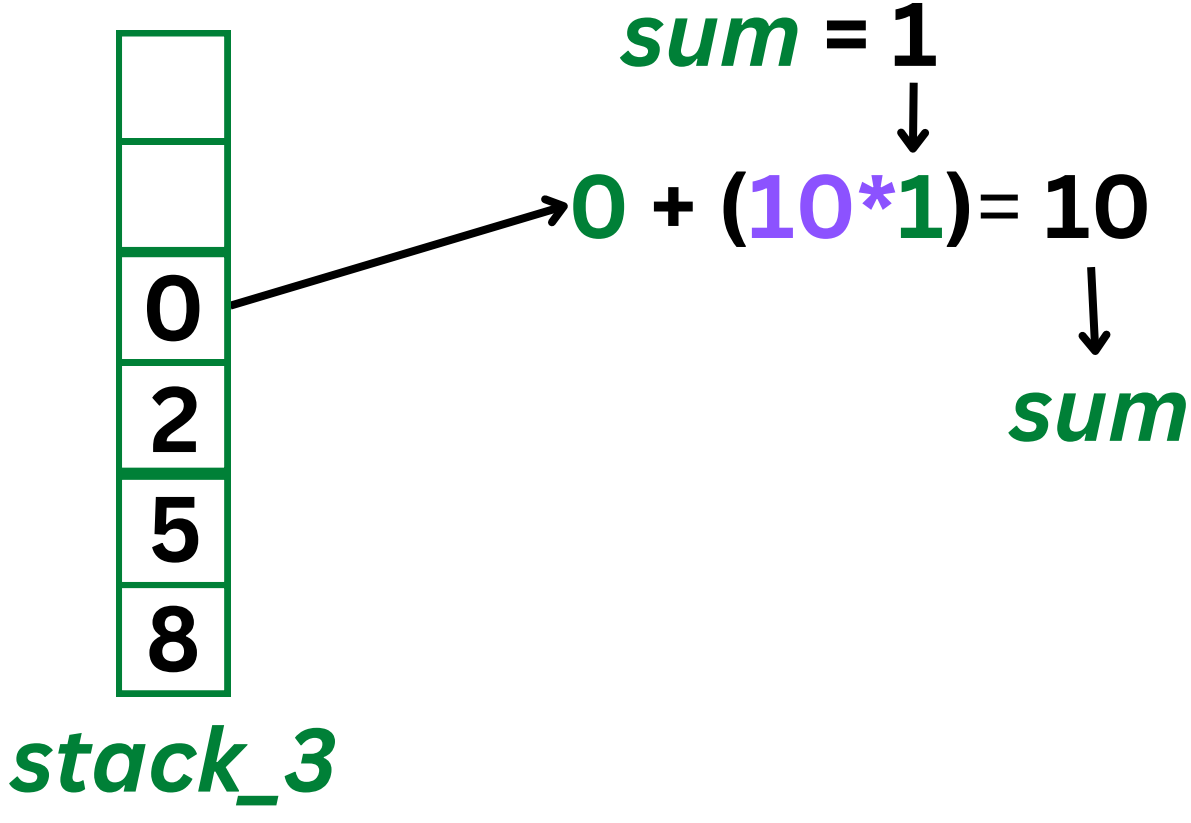
Step 6 loop 1:

Change the value of ***sum*** to 0. Pop from ***stack_3*** and add it to **10 times *sum*** *until the stack is empty*.



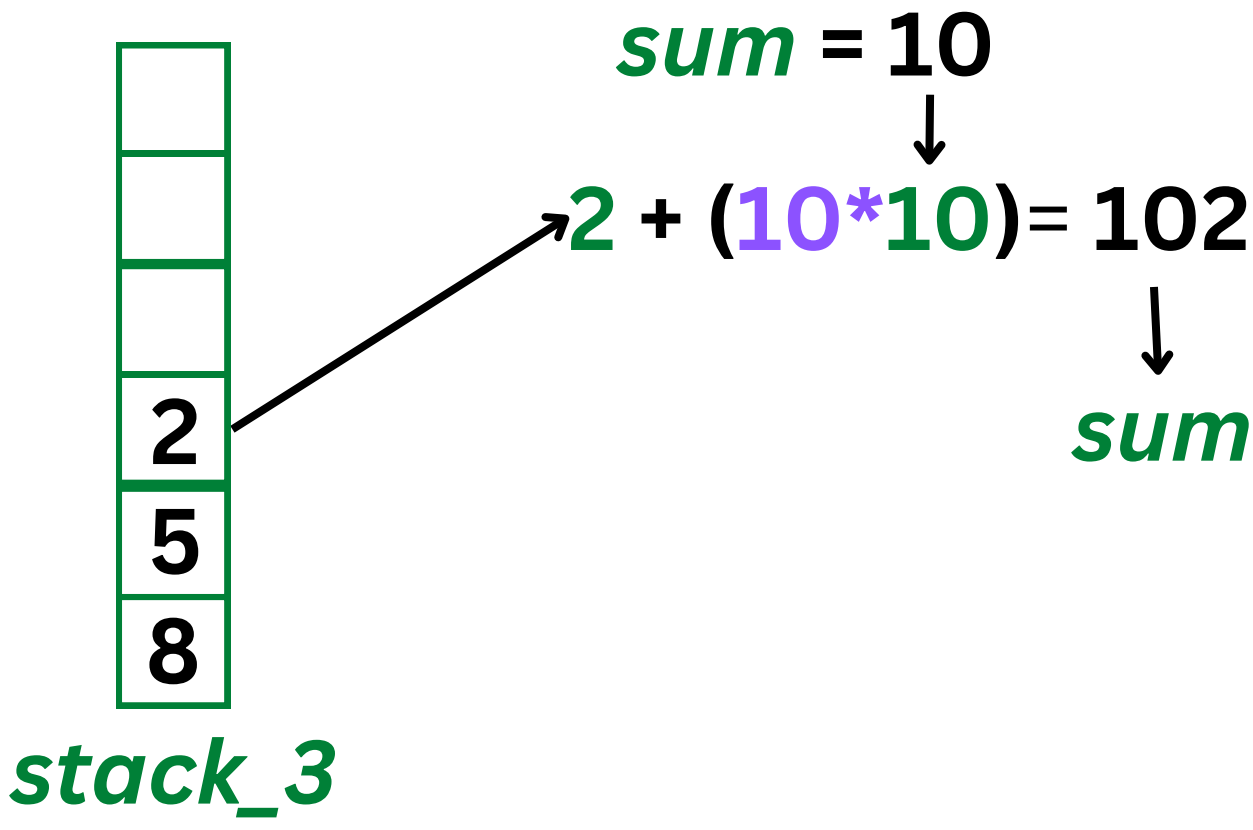
Step 6 loop 2:

Pop from ***stack_3*** and add it to **10 times *sum*** *until the stack is empty.*



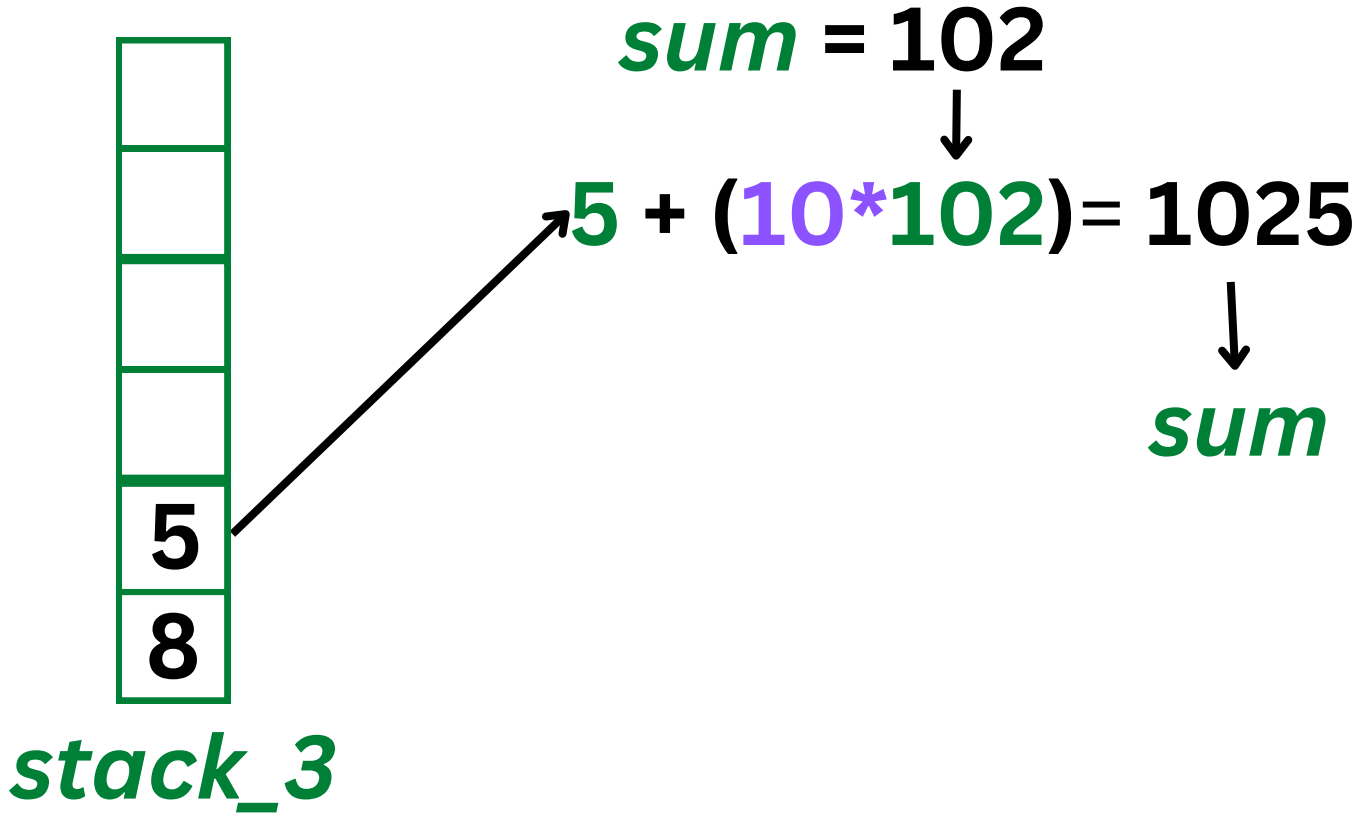
Step 6 loop 3:

Pop from ***stack_3*** and add it to **10 times *sum*** *until the stack is empty*.



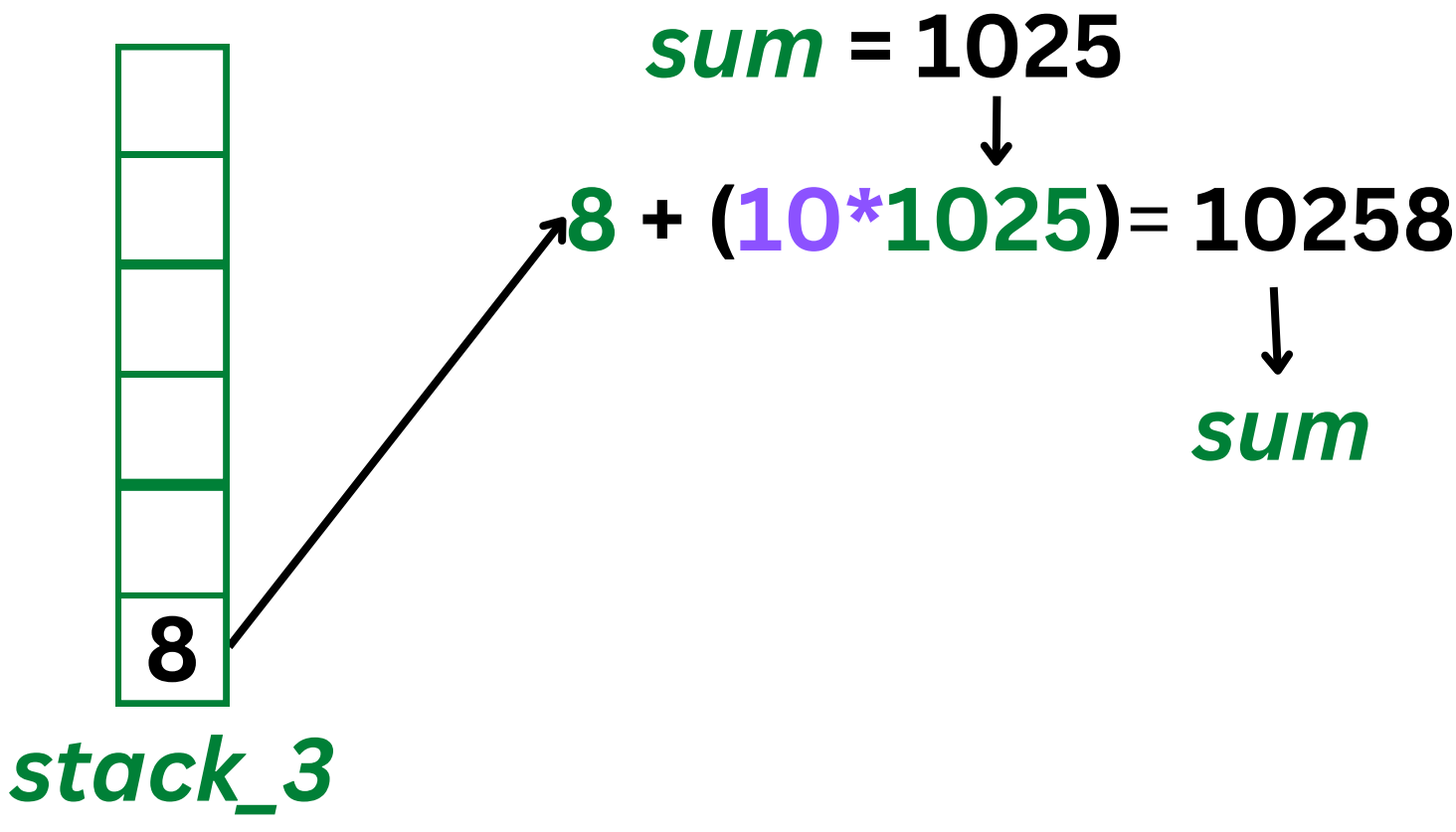
Step 6 loop 4:

Pop from ***stack_3*** and add it to **10 times *sum*** *until the stack is empty.*



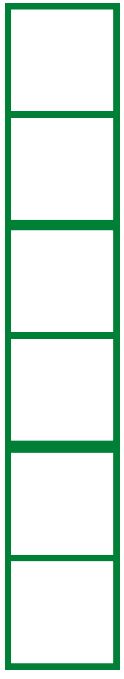
Step 6 loop 5:

Pop from ***stack_3*** and add it to **10 times *sum*** *until the stack is empty.*



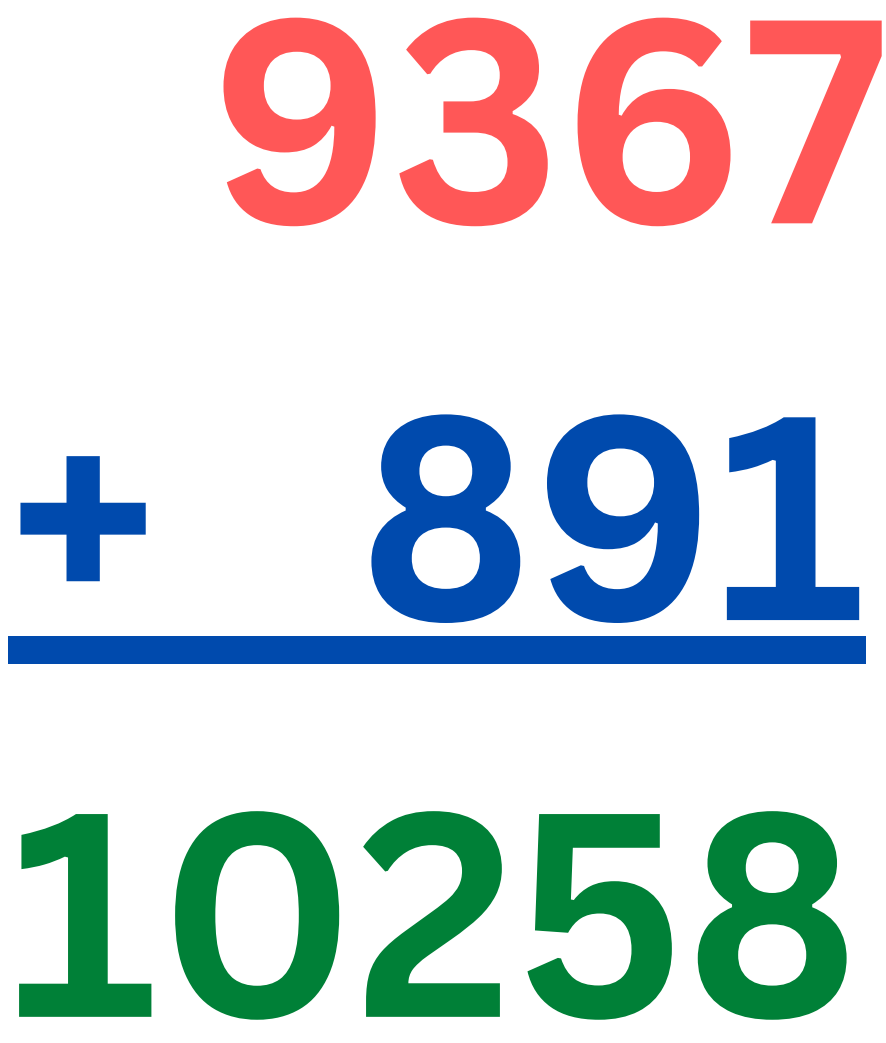
Step 6 end of loop:

Since ***stack_3*** is now empty, we will terminate the loop. We have the final sum which is 10,258.



stack_3

***sum* = 10258**


9367
+ 891
10258