

pseudo code

1/14/21

sum of positives

input  
return the sum  
of all of the  
positive ones  
numbers.



function



output

Example  $[1, -4, 7, 12] \Rightarrow 1 + 7 + 12 = 20$

Note: if there is nothing to sum, the sum is default to 0.

1. set function to positiveSum with  
parameter/argument of arr

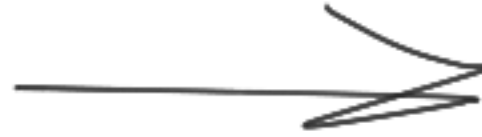


3. declare the variable of i to  
zero, pointing to the first  
value in the array of numbers  
that need to be added. this  
number will increment after  
every loop, by one, pointing to  
every value in the array starting  
at the first value



2. declare the sum variable value to zero, just  
to start off.  
this variable will increment in value while the  
loop is being iterated

4. loop of FOR, statement 1,  $i = 0$  and is executed (one time) before the execution of the code block. Statement 1 sets a value to  $i$  variable before the loop starts ( $i = 0$ ) this points to the first value in the array. Statement 2 defines the condition for the loop to run or execute the code block ( $i$  must be less than `arr.length`).  $i$  must be less, so that way it loops through all the digits in the array, (1,2,3,4,5) and stops at the last digit of the array\*/  
Statement 3 increases the value of  $i$  ( $i++$ ) each time the code block in the loop has been executed. it is incremented after the code block is executed.... And in each increment pointing/looping through the next value in the array. in this case it basically goes through each number of the array. it increases in value so it will loop around the next number in the array\*/



if `arr` at the spot that  $i$  has landed is greater than zero. this line has to be true to execute the next line.  
the line will execute



`sum` plus and equals `arr` at  $i$  spot, exp the `sum` value will change through each iteration, and will be added with the value of `arr` at the  $i$  spot on the array. and will be new value of `sum`, until the next loop changes it.....



then return the total though the `sum` variable