## Workshop One Simple Arrays, Flowcharts and Playing Computer

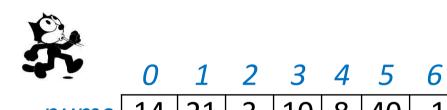
SCC 120 Introduction to Data Structures

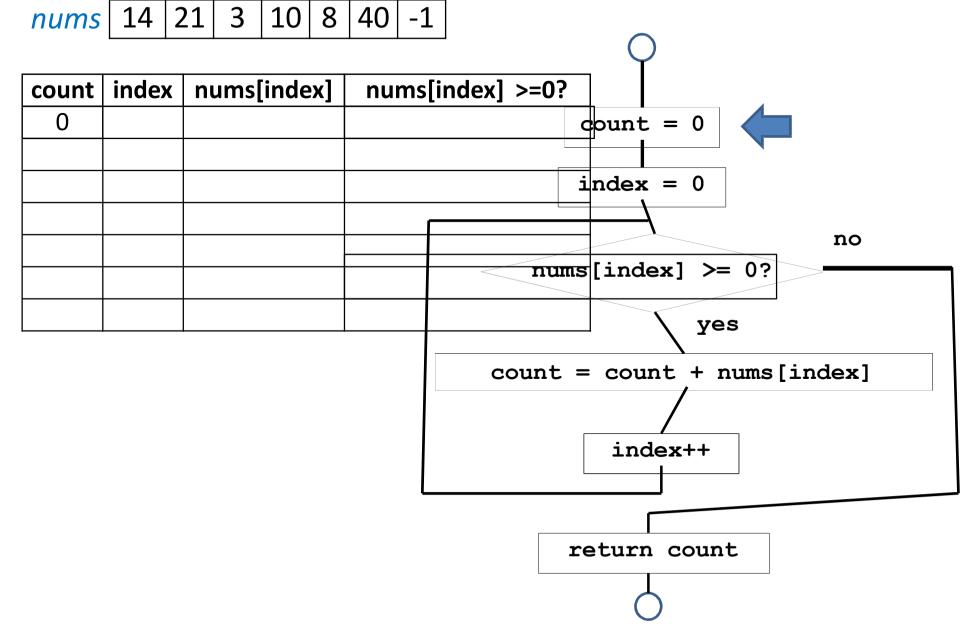
Q1: -14

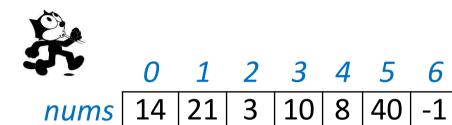
Q2: 11101100

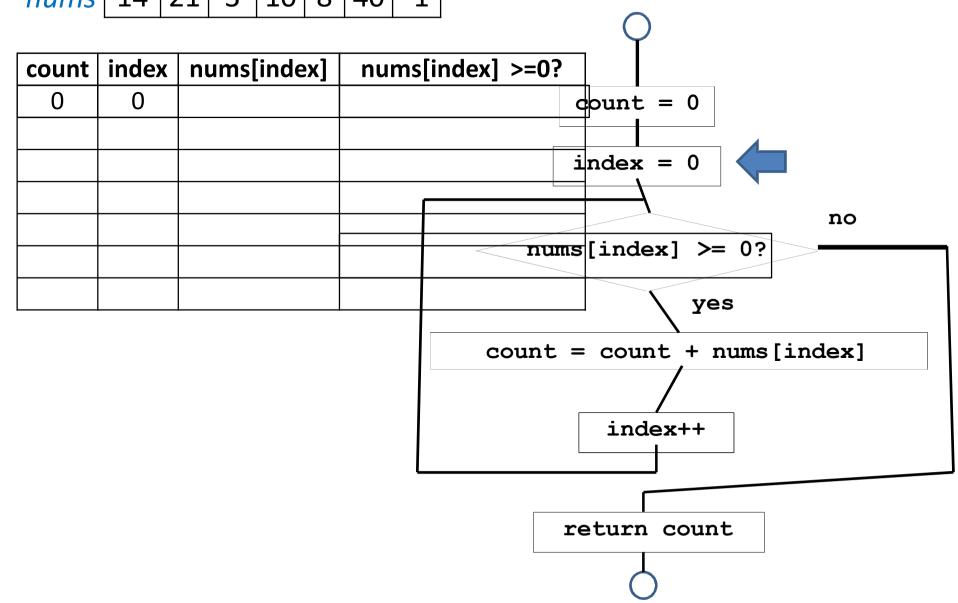
Q3

## EXERCISE ONE. WORKED EXAMPLE OF PLAYING COMPUTER



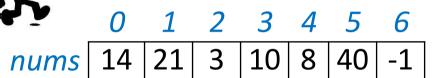








14 >= 0? yes



	<u> </u>		
count	index	nums[index]	nums[index] >=0?
0	0	14	yes count = 0
			index = 0
			no
			nums[index] >= 0?
			yes
	•		
			count = count + nums[index]
			index++
nums[i	indexl	>= 0?	
index =	_	•	
			return count
nums[	J] = 14	•	



= 14

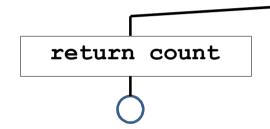


	LL_		$\bigcirc$
count	index	nums[index]	nums[index] >=0?
0	0	14	yes count = 0
14			
			index = 0
			no
			nums[index] >= 0?
			yes
			count = count + nums[index]
			index++
count :	= coun	t + nums[ind	dex1
= 0 + n		_	
	_	J	return count
= 0 + 1	4		



0	1	2	3	4	5	6
14	21	3	10	8	40	-1

				$\mathbf{Q}$
count	index	nums[index]	nums[index] >=0?	
0	0	14	yes <b>c</b> o	ount = 0
14	1			
			in	dex = 0
				no
			nums[:	index] >= 0?
				yes
			count =	count + nums[index]
				index++



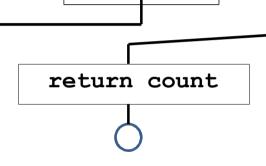


nums

0	1	2	3	4	5	6
14	21	3	10	8	40	-1

Y					
	>=0?	nums[inde	nums[index]	index	count
ount = 0	C	yes	14	0	0
			21	1	14
ndex = 0	i				
no					
[index] >= 0?	nums				
yes					
<pre>count + nums[index]</pre>	ount =				

nums[index] >= 0?
index = 1
nums[1] = 21
21>= 0? yes



index++



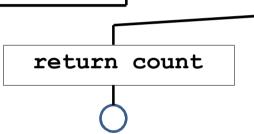
0	1	2	3	4	5	6
14	21	3	10	8	40	-1

$\mathbf{Y}$				
?	nums[index] >=0?	nums[index]	index	count
count = 0	yes	14	0	0
	yes	21	1	14
index = 0				35
ms[index] >=	num			
yes				

count = count + nums[index]

$$= 14 + nums[1]$$

$$= 14 + 21$$



index++

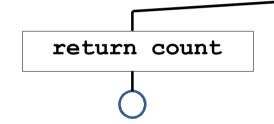
count = count + nums[index]



4 4	24	_	40		40	
0	1	2	3	4	5	6

nums | 14 | 21 | 3 | 10 | 8 | 40 | -1 |

				Q
count	index	nums[index]	nums[index] >	·=0?
0	0	14	yes	count = 0
14	1	21	yes	
35	2			index = 0
				no nums[index] >= 0?  yes
			cou	<pre>unt = count + nums[index]</pre>
				index++





nums

0	1	2	3	4	5	6
14	21	3	10	8	40	-1

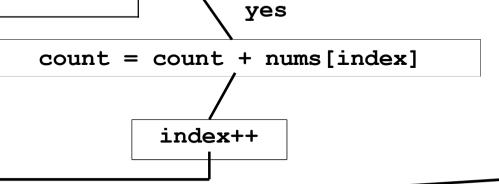
Y					
]	>=0?	nums[index]	nums[index]	index	count
ount = 0	C	yes	14	0	0
		yes	21	1	14
ndex = 0	i		3	2	35
[index1 >=	nums				

nums[index] >= 0?

index = 2

nums[2] = 3

3>= 0? yes



return count



0	1	2	3	4	5	6
14	21	3	10	8	40	-1

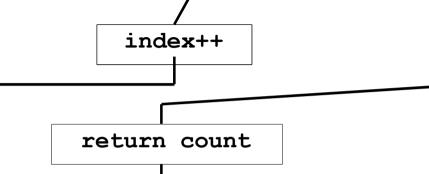
	Y				
		nums[index] >=0?	nums[index]	index	count
: = 0	ount	yes	14	0	0
		yes	21	1	14
x = 0	ndex	yes	3	2	35
1	<b>\</b>				38
ex] >= 0?	[ind	num			
ves					

count = count + nums[index]

$$= 35 + nums[2]$$

$$= 35 + 3$$

= 38



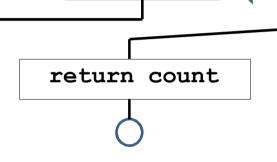
count = count + nums[index]



0	1	2	3	4	5	6
14	21	3	10	8	40	-1

count	index	nums[index]	nums[index] >=0?	
0	0	14	yes	count = 0
14	1	21	yes	
35	2	3	yes	index = 0
38	3			
			num	s[index] >= 0?
				yes

index = index + 1 = 2 + 1 = 3



index++

count = count + nums[index]



nums

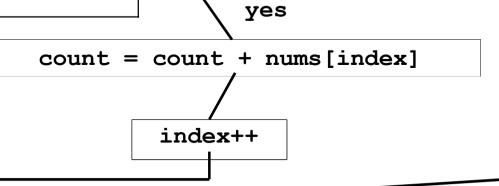
0	1	2	3	4	5	6
14	21	3	10	8	40	-1

T				
	nums[index] >=0?	nums[index]	index	count
ount = 0	yes	14	0	0
	yes	21	1	14
ndex = 0	yes i	3	2	35
		10	3	38
[index] >=	nime			

nums[index] >= 0?
index = 3

nums[3] = 10

10 >= 0? yes



return count



0	1	2	3	4	5	6
1/1	21	2	10	Q	10	1

100 <del>000</del> 0		_			•		
nums	14	21	ო	10	8	40	-1

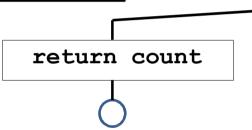
$\vee$					
	?	nums[index] >=0?	nums[index]	index	count
ount = 0	С	yes	14	0	0
		yes	21	1	14
ndex = 0	i	yes	3	2	35
		yes	10	3	38
no					48
index] >= 0?	ıms	nu			
yes 🛑					
			•		
<pre>count + nums[index]</pre>	; =	count			

count = count + nums[index]

$$= 38 + nums[3]$$

$$= 38 + 10$$

= 48



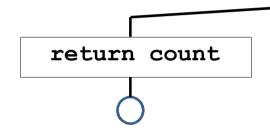
index++



0	1	3	4	5	6

14	21	ന	10	8	40	-1

	•		<del></del>
count	index	nums[index]	nums[index] >=0?
0	0	14	yes count = 0
14	1	21	yes
35	2	3	yes index = 0
38	3	10	yes
48	4		no
			nums [index] >= 0?
			yes
			count = count + nums[index]
			index++



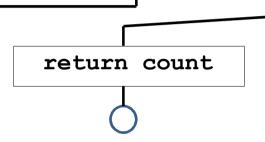


nums

0	1	2	3	4	5	6
14	21	3	10	8	40	-1

<b>,</b>					
		ıms[index] >=0?	nums[index]	index	count
= 0	ount	yes	14	0	0
		yes	21	1	14
= 0	index	yes	3	2	35
		<del>yes</del>	10	3	38
			8	4	48
ex] >= 0?	[ind	num			
yes					

nums[index] >= 0? index = 4 nums[4] = 8 8 >= 0? yes



index++

count = count + nums[index]



0	1	2	3	4	5	6
14	21	3	10	8	40	-1

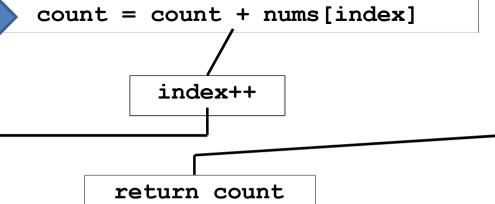
count	index	nums[index]	nums[index] >=0?	$\neg$
0	0	14	yes	count = 0
14	1	21	yes	
35	2	3	yes	index = 0
38	3	10	<del>r yes</del>	
48	4	8	yes	no
56			nun	is[index] >= 0?
				yes 🛑

count = count + nums[index]

$$= 48 + nums[4]$$

= 48+ 8

= 56





56

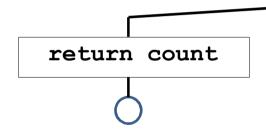
5

	0						
nums	14	21	3	10	8	40	-1

,	Y				
		nums[index] >=0?	nums[index]	index	count
= 0	ount =	yes	14	0	0
		yes	21	1	14
= 0	ndex =	yes	3	2	35
		yes	10	3	38
		yes	8	4	48
=x] >= 0	index	num			56

yes count = count + nums[index] index++

index = index + 1= 4 + 1= 5



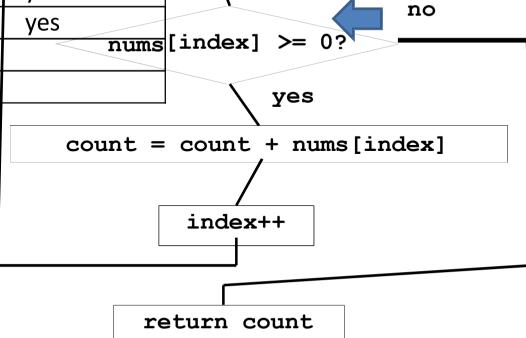


nums

0	1	2	3	4	5	6
14	21	3	10	8	40	-1

ı I			!l	
	s[index] >=0?	nums[index]	inaex	count
ount = 0	yes	14	0	0
	yes	21	1	14
ndex = 0	yes i	3	2	35
	yes	10	3	38
	yes	8	4	48
[index] >= 0	nums	40	5	56

nums[index] >= 0? index = 5 nums[5] = 40 40 >= 0? yes





0	1	2	3	4	5	6
1 /	21	7	10	)	10	1

nums	14	21	ന	10	8	40	-1

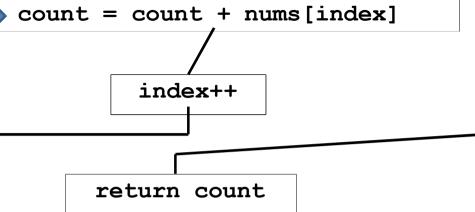
count	index	nums[index]	nums[index] >=0?			
0	0	14	yes	count	= 0	
14	1	21	yes			
35	2	3	yes	index	= 0	
38	3	10	yes			
48	4	8	yes	,	•	no
56	5	40	yes	ਤਿ[inde	ex] >=	= 0?
96					ve	es 👍

count = count + nums[index]

= 56 + nums[5]

= 56 + 40

= 96





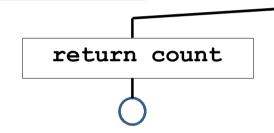
4 4						
0	1	2	3	4	5	6

nums

14	21	3	10	8	40	-1

				Y
count	index	nums[index]	nums[index] >=0?	
0	0	14	yes	count = 0
14	1	21	yes	
35	2	3	yes	index = 0
38	3	10	<del>yes</del>	
48	4	8	yes	no
56	5	40	yes	s[index] >= 0?
96	6			yes
			count :	= count + nums[index]

index = index + 1 = 5 + 1 = 6



index++

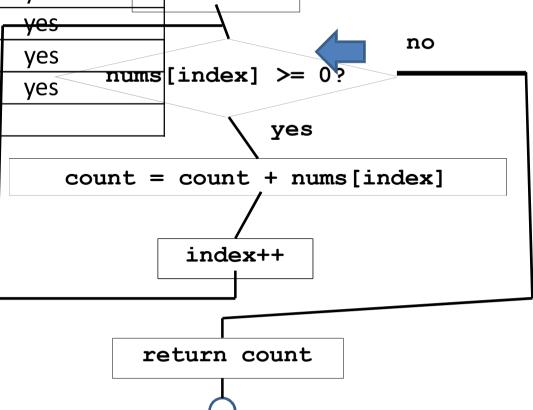


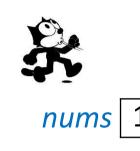
nums

0	1	2	3	4	5	6
14	21	3	10	8	40	-1

		[index] >=0?	nums	nums[index]	index	count
t = 0	coun	yes		14	0	0
		yes		21	1	14
к = 0	inde	yes		3	2	35
7		yes		10	3	38
		yes		8	4	48
lex] >=	Blivo	yes		40	5	56
yes				-1	6	96

nums[index] >= 0? index = 6 nums[6] = -1 -1 >= 0? no





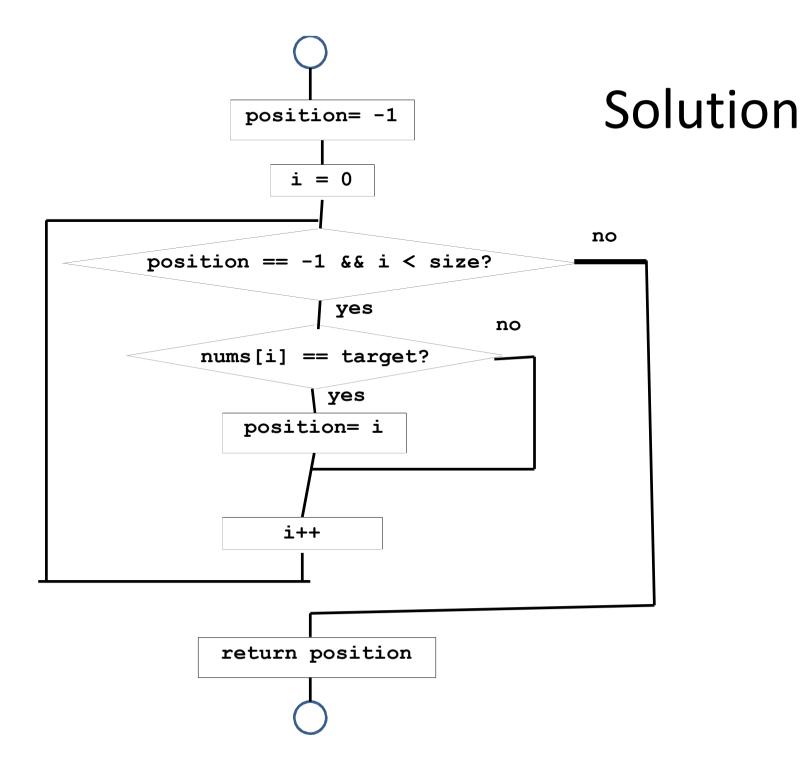
0						
14	21	3	10	8	40	-1

count	index	nums[index]	nums[index] >=0?
0	0	14	yes count = 0
14	1	21	yes
35	2	3	yes index = 0
38	3	10	yes
48	4	8	yes
56	5	40	yes nums [index] >= 0?
96	6	-1	no yes
			<pre>count = count + nums[index] index++</pre>
count -	- 06		return count

count = 96

## **EXERCISE TWO**

- Draw a flowchart for the "search" function.
- The signature for this function has three parameters: the array of integers to be searched; the "size" of the array really, the number of elements to be searched; and finally, the target value we are looking for. The function either returns the position of the target value, if it is present, or -1 to indicate that the target value is not present.



## The End