Understanding the Problem

The main goal of the program is making linked list class and running the function in the linked list class in the main function. The program will sort the input values including 0, positive numbers, and negative numbers by ascending order or descending order. Furthermore, the program will find the prime numbers among the input values, and then count the numbers of the prime numbers. In the process of sorting numbers, the program will swap the nodes which is pointing the input value, not just the input value.

<Assumption>

I assume that the program will not regard the negative numbers as a prime number.

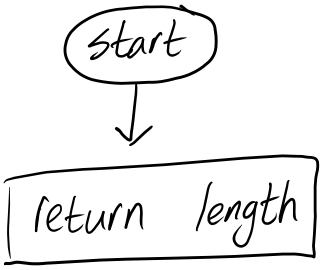
I assume that the program will implement the user's request completely.

I assume that the program will have some recursion functions.

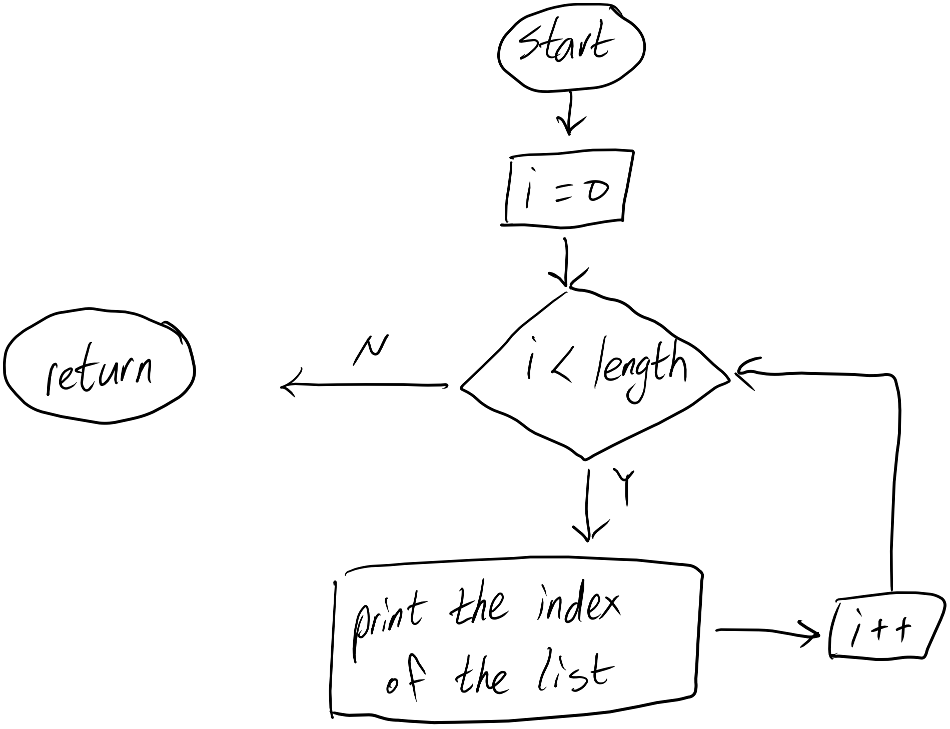
I assume that if there are same numbers in the array, the program will consider them when the program is going to sort the numbers of the array.

Program Design

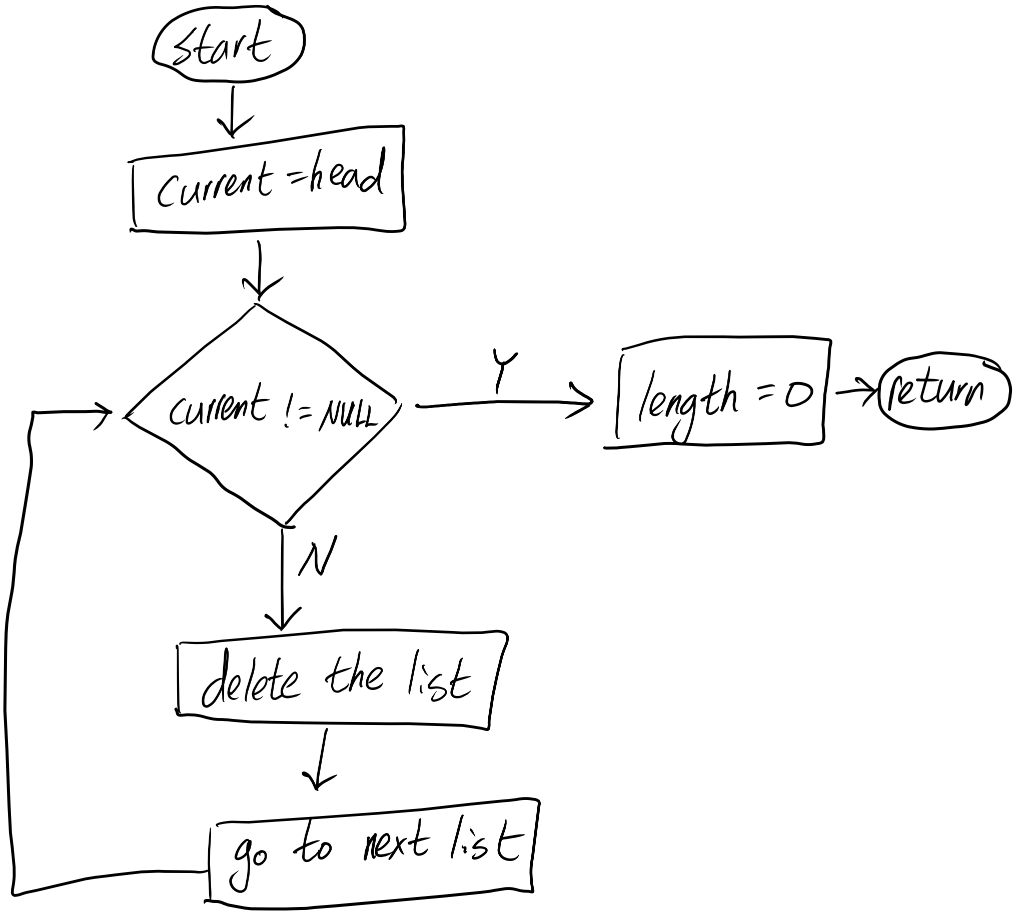
<get_length( )>



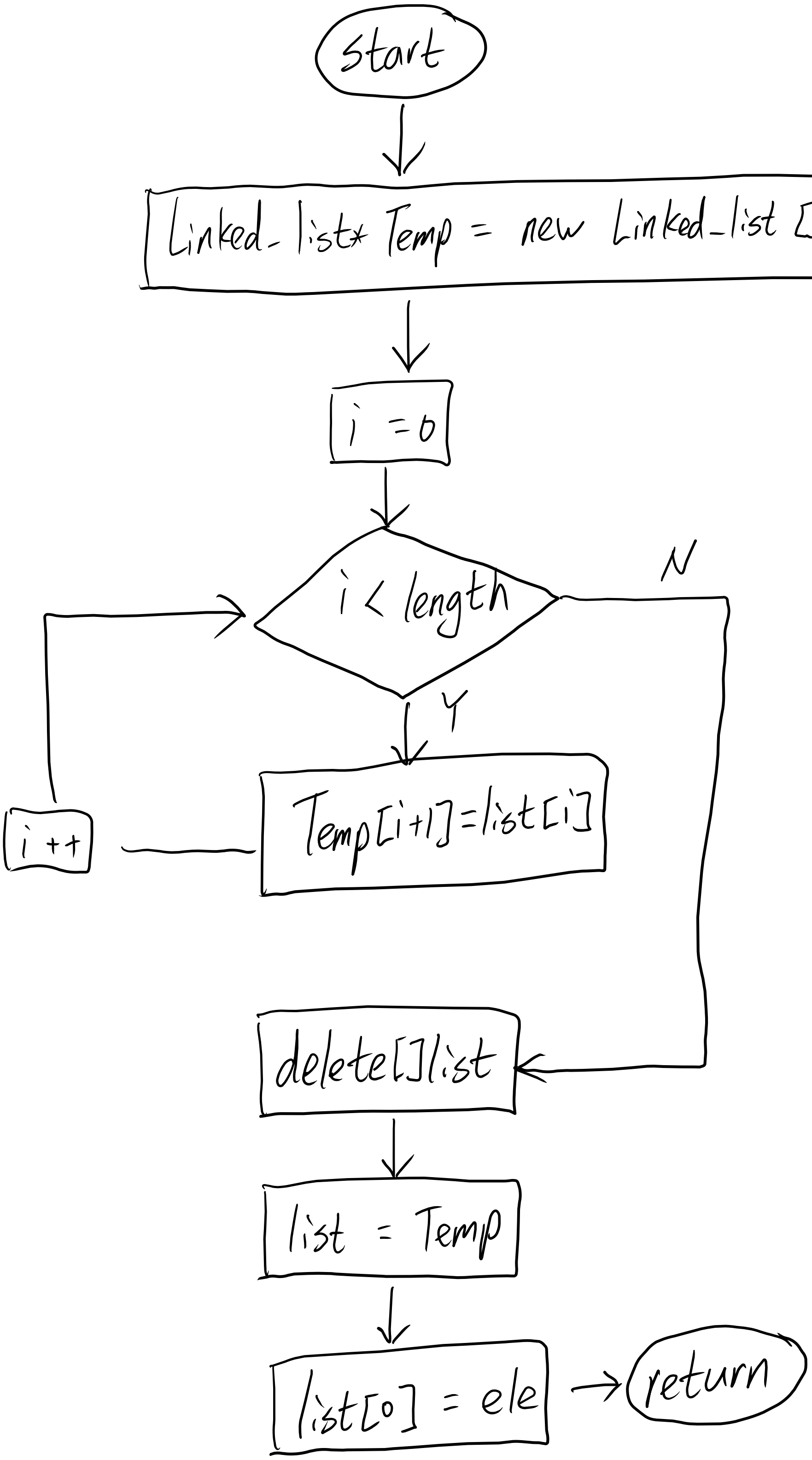
<print( )>



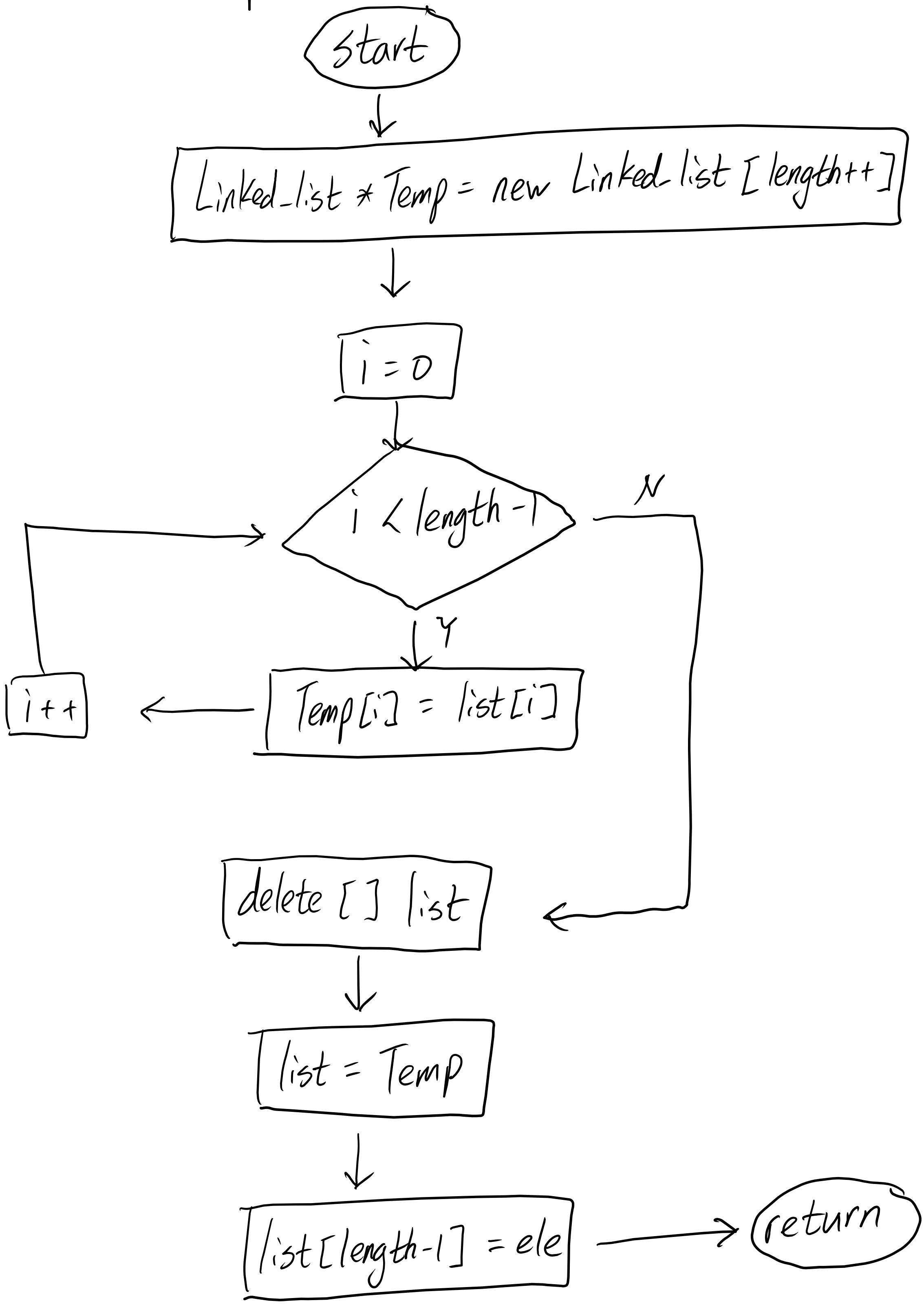
<clear( )>


start
current = head
current != NULL
Y
length = 0
return
N
delete the list
go to next list

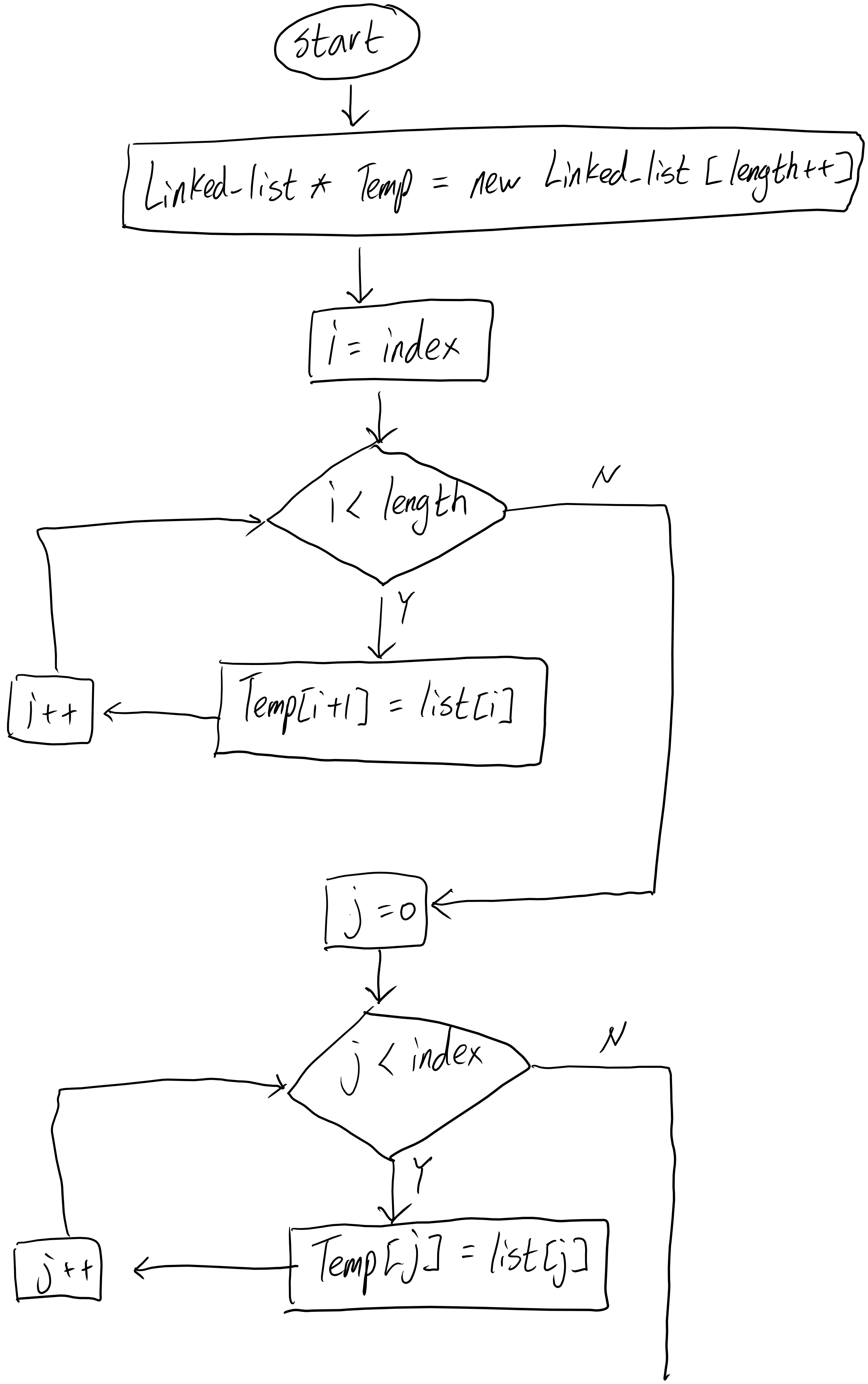
<push_front(ele)>

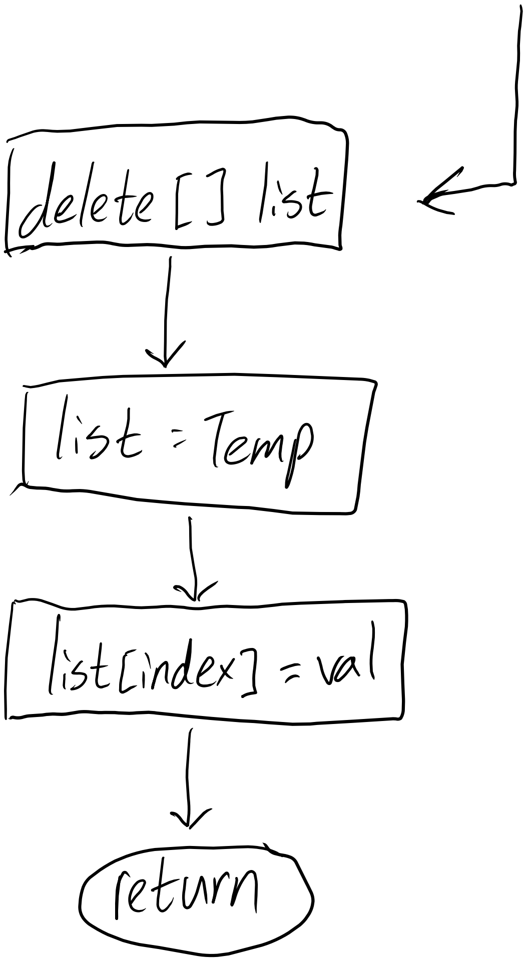

start
Linked_list* Temp = new Linked_list [length++]
i = 0
i < length
N
Y
Temp[i+1]=list[i]
i ++
delete[]list
list = Temp
list[0] = ele
return

<push_back(ele)>

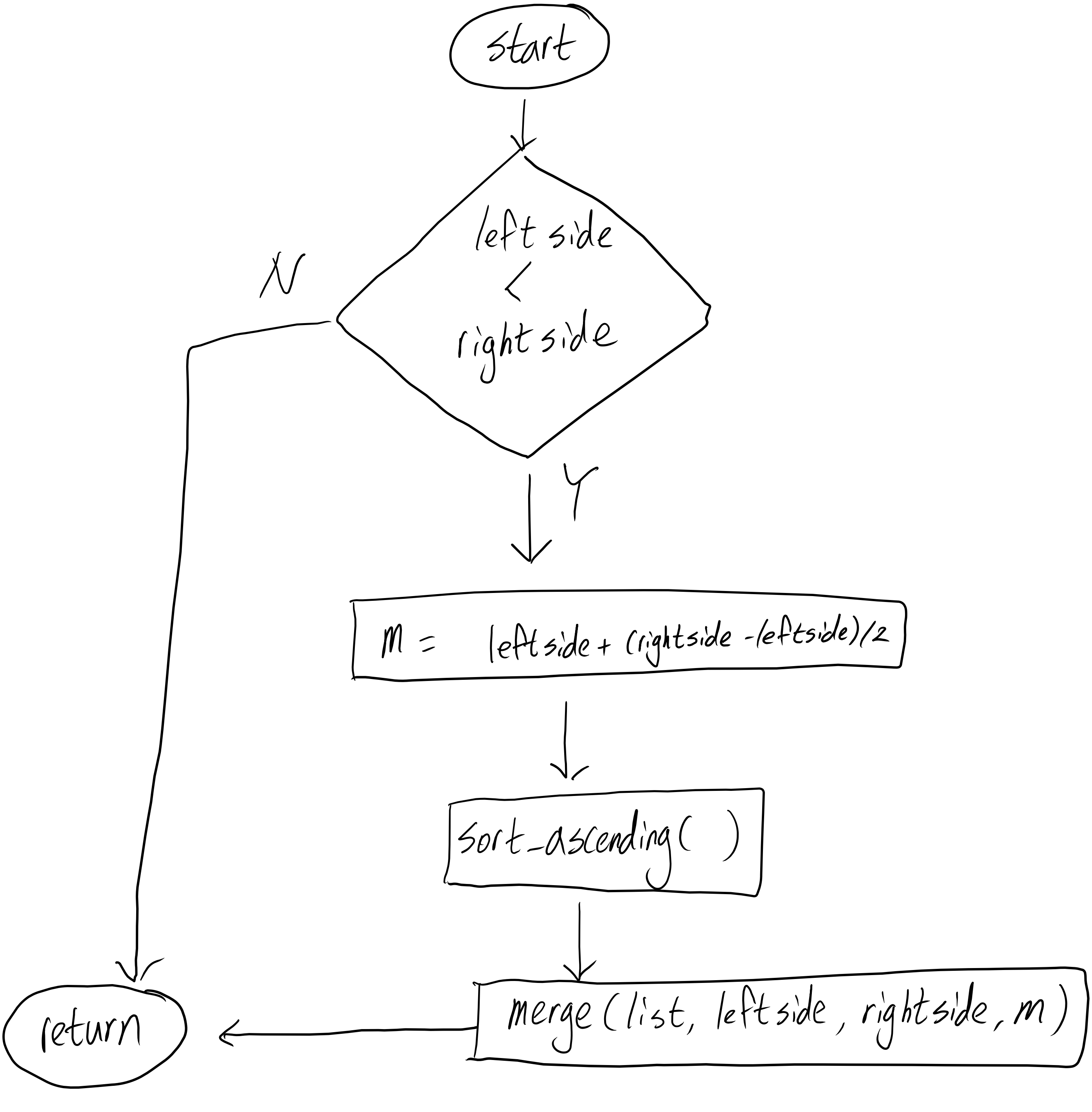

Start
Linked_list * Temp = new Linked_list [length++]
i = 0
i < length-1
N
Y
Temp[i] = list[i]
i++
delete [] list
list = Temp
list[length-1] = ele
return

< insert(val, index) >

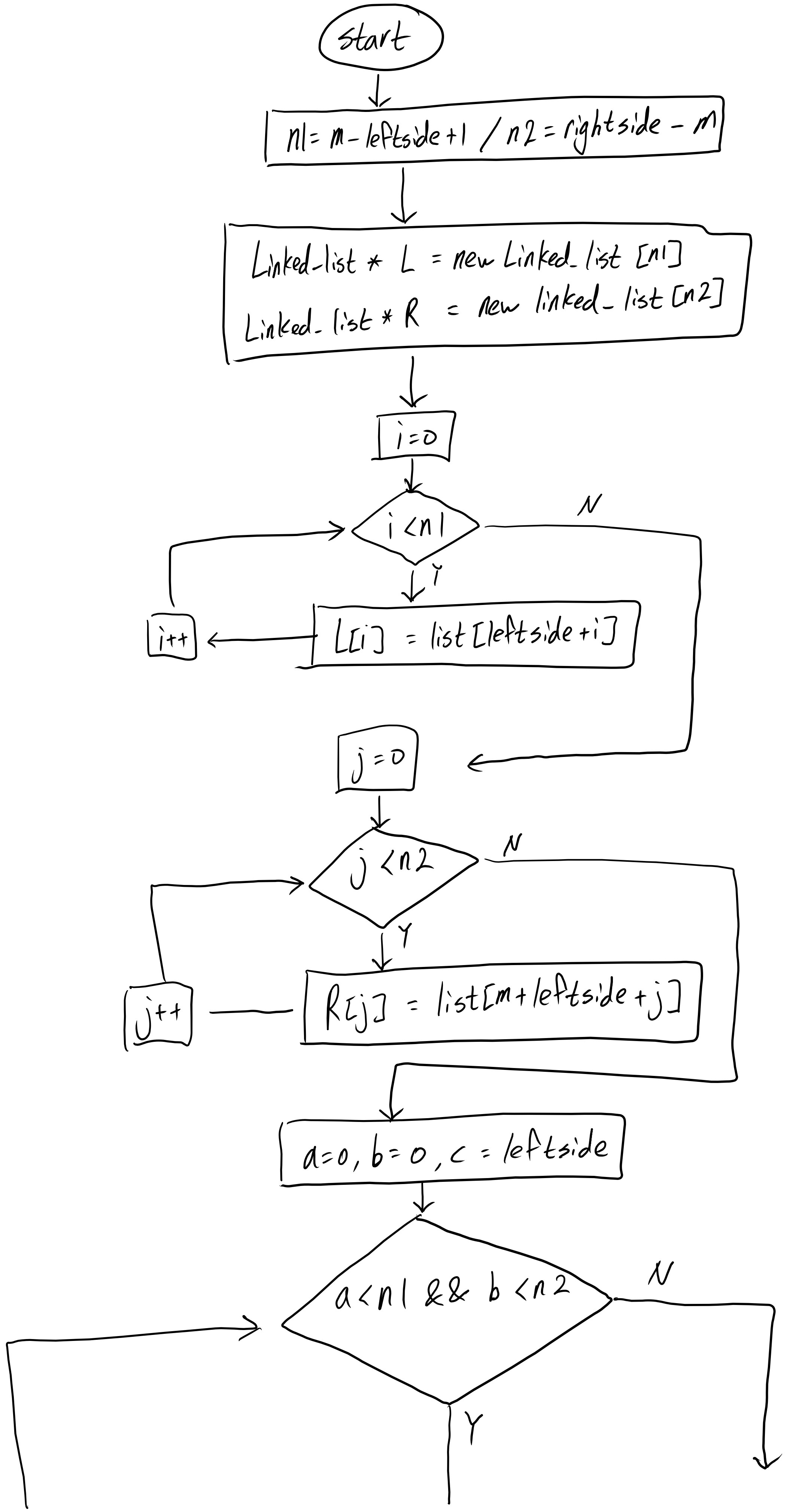


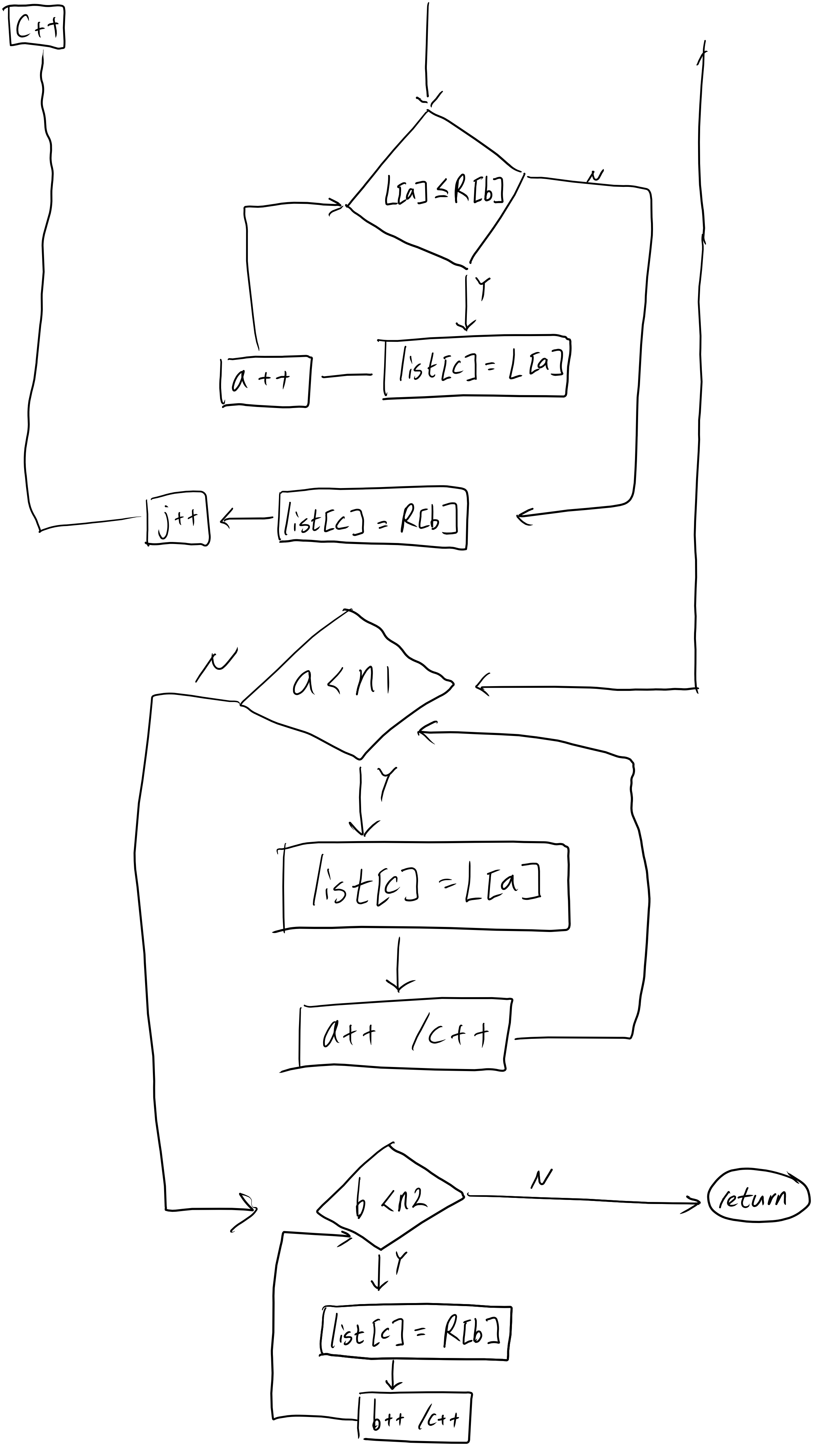

delete[] list
list = Temp
list[index] = val
return

< sort_ascending( ) >

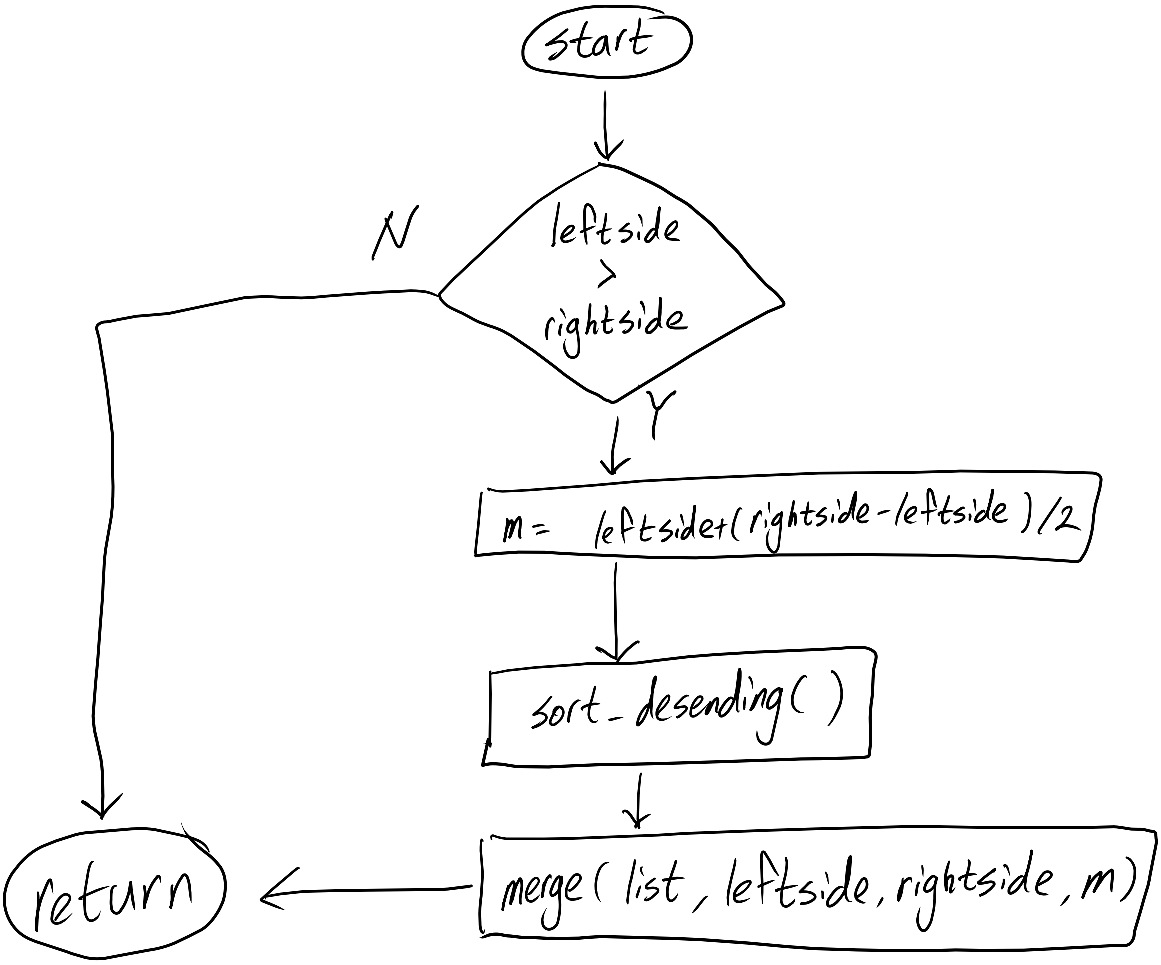

start
left side
<
right side
N
Y
m = leftside + (rightside -leftside)/2
sort_ascending( )
merge(list, leftside, rightside, m)
return

<merge(list, leftside, rightside, m)>

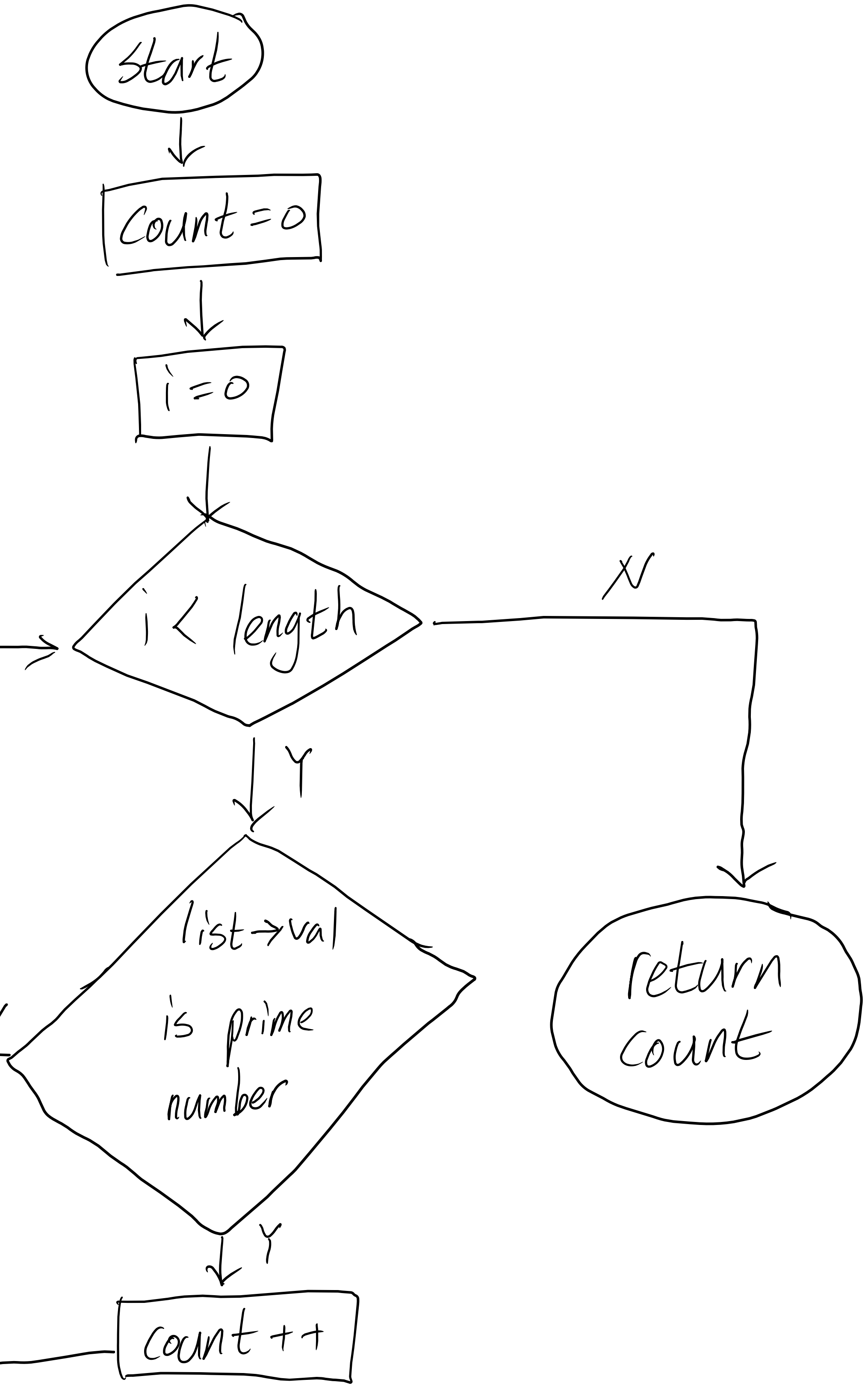

start
n1= m-leftside+1 / n2=rightside - m
Linked-list * L = new Linked_list [n1]
Linked_list * R = new linked_list [n2]
i=0
i<n1
N
Y
L[i] = list[leftside+i]
i++
j=0
j<n2
N
Y
R[j] = list[m+leftside+j]
j++
a=0, b=0, c = leftside
a<n1 && b<n2
N
Y


C++
L[a] ≤ R[b]
N
Y
list[c] = L[a]
a ++
list[c] = R[b]
j++
N
a < n1
Y
list[c] = L[a]
a++ /c++
b < n2
N
return
Y
list[c] = R[b]
b++ /c++

< sort_desending( )>


start
leftside > rightside
N
Y
m= leftside+(rightside-leftside)/2
sort_desending( )
merge(list, leftside, rightside, m)
return

< count_prime(const linked_list list)



Program Testing

Program Setting	Input	Expected Result
<case 1>	List.push_front(0)	[0, 1, -1, 0, 5, 9, 3]
<case 1>	List.push_back(10)	[1, -1, 0, 5, 9, 3, 10]
<case 2>	List.sort_ascending()	[-1, 0, 0, 0, 1, 1]
<case 2>	List.sort_desending()	[1, 1, 0, 0, 0, -1]
<case 1>	List.clear()	[ ]
<case 2>	List.insert(2, 3)	[0, 0, 0, 2, -1, 1, 1]

<case 1>

[1, -1, 0, 5, 9, 3]

<case 2>

[0, 0, 0, -1, 1, 1]