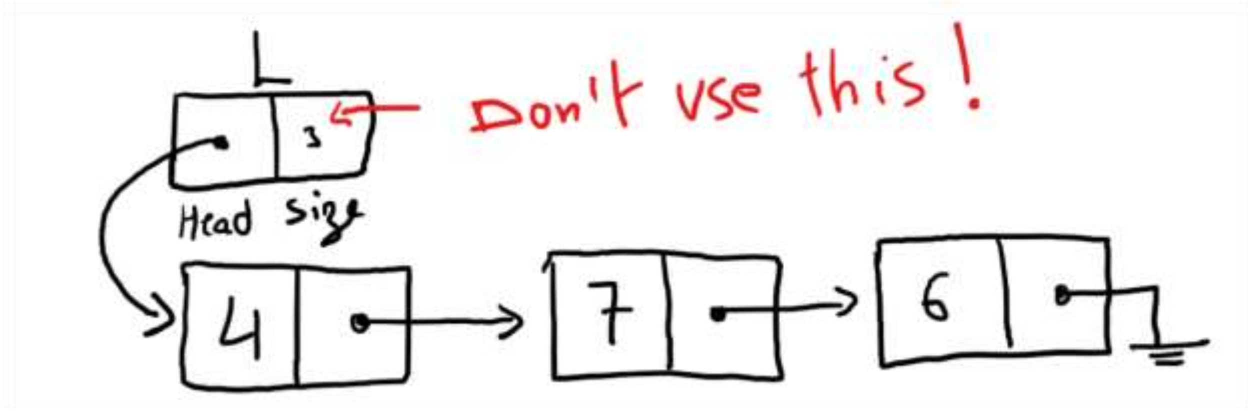
Problem 1:

Write a function int size_list(list * l) which given a linked list l , it calculate and returns the number of elements inside the list.

Note : In these problem , do not use the size property of the list_node , in other word, iterate through all the elements of the list and count the number of nodes.

Bonus: Solve this problem using iterative and recursive approaches

Sample input:



Sample Output:

```
3
```

Problem 2:

Write a function list_node * find_list(list * l, int value) which given a linked list and a value, it searches for the value inside the list and returns the address of its list_node , or NULL of the value was not found.

Bonus: Solve this problem using iterative and recursive approaches.

Problem 3:

Write a function list_node * find_list(list * l, int pos) which given a linked list and a position , it returns the address of the node number pos inside the list, or NULL if pos is bigger than the size of the node was not found.
Note : the pos of the first node is 1 , not 0.

Bonus: Solve this problem using iterative and recursive approaches.

Problem 4:

Write a function void clear_list(list * l) which given a linked list l , it removes all the list_nodes of the list.

Problem 5:

Write a function void selection_sort(list * l) which sorts a given list using bubble sort algorithm.