Question - 1

Union Operation

SCORE: **5 points**

How many connected components result after performing the following sequence of union operations on a set of 10 items?

```
1-2  3-4  5-6  7-8  7-9  2-8  0-5  1-9
```

Assume an array of size of 10, like below.

```
[0 1 2 3 4 5 6 7 8 9 ]
```

- 4
- 3
- 1
- 5

Question - 2

Quick Find

SCORE: **5 points**

Statement 1: Quick-Find union operation is too expensive
Statement 2: Trees formed in Quick-Union are always flat
Statement 3: Find / connected operation can be N-array access in Quick-Union, hence it is too expensive
Statement 4: It takes O(N) array accesses to process one union operation on N objects in Quick-Find
Which statements are true?

- 1
- 2
- 3
- 4
- None of the above

Question - 3

ADT

SCORE: **5 points**

Help

A data *buffer* is a region to temporarily store data while it is being moved from one place to another, Typically, the data is stored in a buffer as it is retrieved from an input device or just before it is sent to an output device. which of the following data structures is best suited to being used as a buffer?

- ( ) Stack
- (•) Queue
- ( ) Bag
- ( ) ArrayList

Question - 4

Stirling Notation

SCORE: **5 points**

What is Stirling's approximation for n! in tilde notation?

- ( ) ~ ln(n)
- ( ) ~ lg(n)
- ( ) ~n ln(n)
- (•) ~n lg(n)

Question - 5

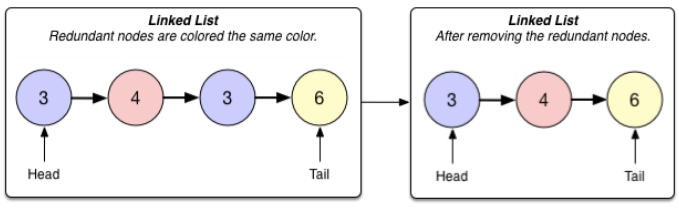
Condensed List

SCORE: **50 points**

Linked Lists | Easy | Data Structures | Algorithms | Problem Solving | Core Skills

Given a list of integers, remove any nodes that have values that have previously occurred in the list and return a reference to the head of the list.

For example, the following list has a recurrence of the value *3* initially:



Remove the node at position *2* in the list above, *0* based indexing.

Function Description

Complete the function *condense* in the editor below. The function must return a reference to a *LinkedListNode*, the first node of a list that

contains only the unique value nodes from the original list, in order.

condense has the following parameter(s):

head: the head of a singly-linked list of integers, a LinkedListNode

Note:. A *LinkedListNode* has two attributes: *val*, an integer, and *next*, a reference to the next item in the list or the language equivalent of *null* at the tail.

Constraints

- $1 \le n \le 10^5$
- $0 \le LinkedListNode[i].val \le 1000$

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *list*.
Each of the next *n* lines contains an integer *list[i]* where $0 \le i < n$.

▼ Sample Case 0

Sample Input 0

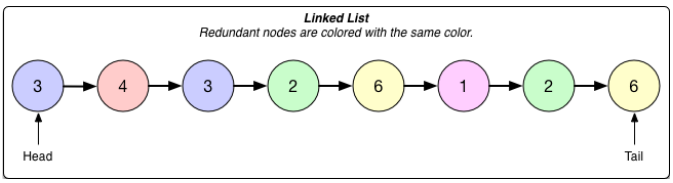
```
8
3
4
3
2
6
1
2
6
```

Sample Output 0

```
3
4
2
6
1
```

Explanation 0

The list looks like this:

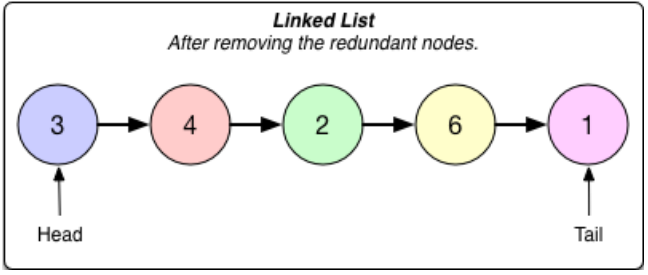


The linked list given as input.

From the diagram, remove:

- *list[2] = 3*
- *list[6] = 2*
- *list[7] = 6*

After calling *condense*, the list looks like this:



The condensed list.