

```

//  ASSIGNMENT 3
//  FENCHAO DU
//
//  main.cpp
//  ASSIGNMENT 3
//
//  Created by Fenchao Du on 7/19/18.
//  Copyright © 2018 Fenchao Du. All rights reserved.
//

#include <iostream>

using namespace std;

struct node_t {
    struct node_t * next;
    struct node_t * next_highest;
    struct node_t * next_higher;
    int val;
};

/* Part 1 start */
// Problem 1
// Time = O(n), Memory = O(n)
// Recursive solution.
// We traverse linked-list till end and reversely return highest or current node.
// Because of recursion, we call function for all n does till end. space = O (n)
// @head : node that we want to set next highest after it
// @return : If current node's val is greater than next highest, return current
//           node.
//           Otherwise, return next highest value of this node
struct node_t * setNextHighest (struct node_t * head) {
    if (!head) {
        return nullptr;
    }
    head->next_highest = setNextHighest(head->next);
    if (head->next_highest && head->next_highest->val > head->val) {
        return head->next_highest;
    }
    return head;
}

/* Part 1 end */

/* Part 2 start */
// Problem 2
// Time = O(n^2), Memory = O(1)
// We traverse all n nodes. For worst case,
// we will check all following n-1 nodes for 0 th node. etc.
// @ head : node that we want to set next higher after it
// @ val : target value
struct node_t * getNextHigher(struct node_t * head, int val) {
    while (head) {

```

```

        if (head->val > val) {
            return head;
        }
        head = head->next;
    }
    return nullptr;
}

// @ head: start of linked list. traverse and set all next higher
void setNextHigher(struct node_t * head) {
    while(head) {
        head->next_higher = getNextHigher(head->next, head->val);
        head = head->next;
    }
}

/* Part 2 end */

```

```

/* TEST Begin */
int main(int argc, const char * argv[]) {
    // insert code here...
    struct node_t * head = new struct node_t;
    struct node_t * tmp = head;
    for (int i = 0; i < 10; ++i) {
        tmp->next_highest = nullptr;
        tmp->next_higher = nullptr;
        tmp->val = i;
        tmp->next = new struct node_t;
        tmp = tmp->next;
    }
    tmp->val = 10;
    tmp->next = nullptr;
    tmp->next_highest = nullptr;
    tmp->next_higher = nullptr;

    setNextHighest(head);
    setNextHigher(head);

    tmp = head;
    while(tmp && tmp->next_highest) {
        cout << tmp->next_highest->val << endl;
        tmp = tmp->next;
    }

    tmp = head;
    while(tmp && tmp->next_higher) {
        cout << tmp->next_higher->val << endl;
        tmp = tmp->next;
    }
    return 0;
}

```