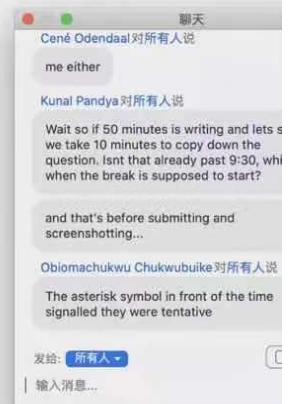
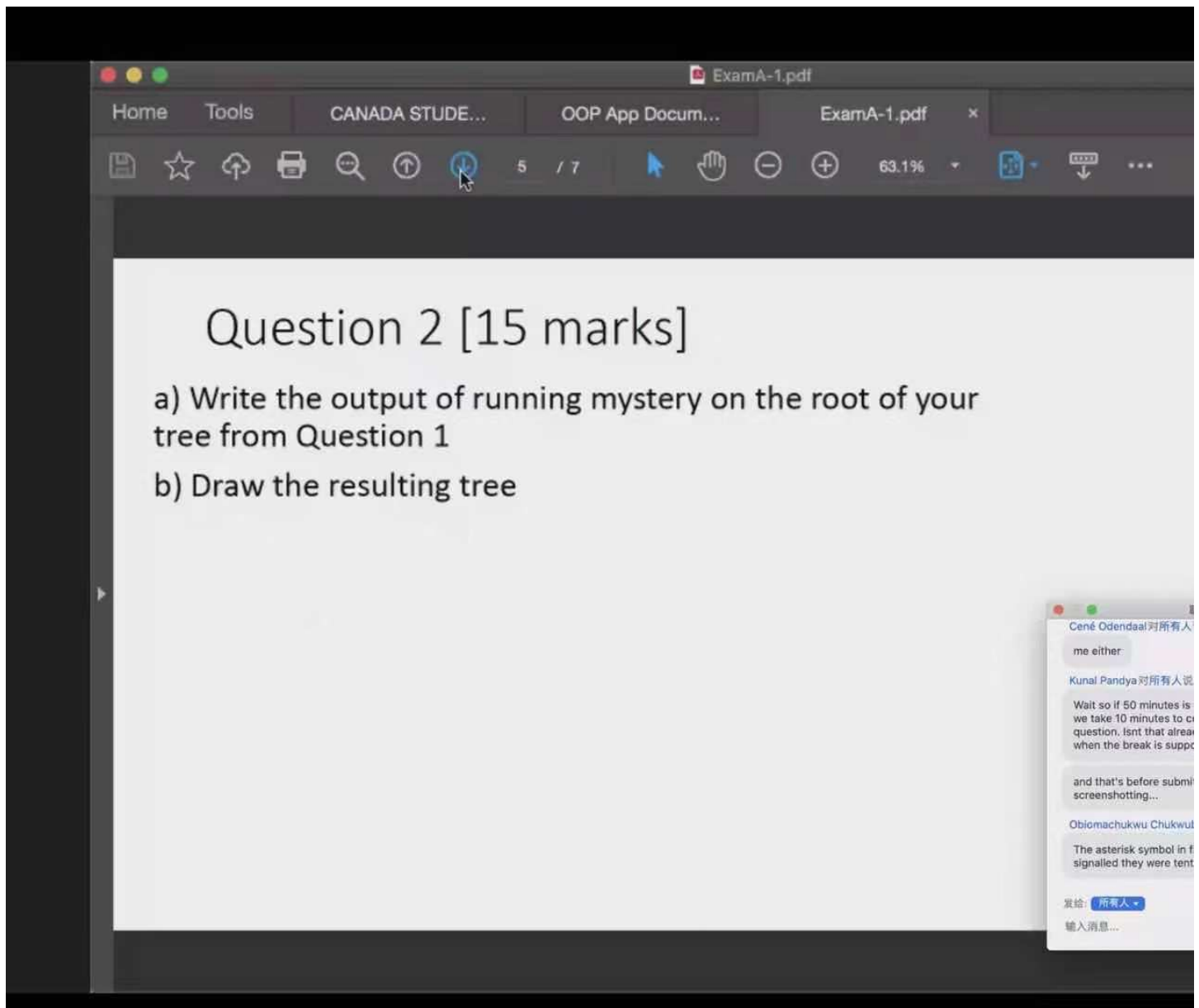


```
Node* mystery(Node* r){
    Node *result = r;
    if(r != NULL){
        r -> left = mystery(r -> left);
        r -> right = mystery(r -> right);
        int b = get_height(r -> left) - get_height(r -> right);
        printf("%d, %d\n", r -> data, b);
        if( b > 1){
            result = r -> left;
            r -> left = r -> left -> right;
            result -> right = r;
        }else if(b < -1){
            result = r -> right;
            r -> right = r -> right -> left;
            result -> left = r;
        }
    }
    return result;
}
```



Question 1:

Draw a BST using your student number, insert from the last digit to the first digit



Question 3 [8 marks]

Give the worst case big-O complexity of mystery being run on the following types of trees where n is the number of nodes in the tree:

- a. A binary tree
- b. A binary search tree
- c. A balanced binary tree
- d. A balanced binary search tree

You only need to give big-O complexities for each, but partial marks may be given for incorrect answers with justifications

聊天

Cené Odendaal 对所有人说

me either

Kunal Pandya 对所有人说

Wait so if 50 minutes is writing and we take 10 minutes to copy down question. Isn't that already past 9 when the break is supposed to start

and that's before submitting and screenshotting...

Oblomachukwu Chukwubike 对所有人说

The asterisk symbol in front of the question signalled they were tentative

Question 4 [3 marks]

Provide a better name for mystery and a brief function description (2-3 sentences max)

聊天

Cené Odendaal 对所有人说

me either

Kunal Pandya 对所有人说

Wait so if 50 minutes is writing we take 10 minutes to copy down question. Isn't that already past when the break is supposed to

and that's before submitting an screenshotting...

Obiomachukwu Chukwubike 对

The asterisk symbol in front of signalled they were tentative

Question 1 [15 marks]

A BackList is a linked list of nodes where each node links to the next node in the list, as well as the previous node in the list that shares its value. The struct for the node is given below.

Write a function insert which takes 3 parameters: the head of a BackList h, a char c and an int i, and inserts c into h at index i

```
typedef struct BackListNode{
    char data;
    //the next node in the BackList
    //NULL if no such node exists
    struct BackListNode *next;
    //the previous node in the BackList
    //with the same data value as this node
    //NULL if no such node exists
    struct BackListNode *prev;
}Node;
```

Question 2 [12 marks]

Write the following functions. (note: t is the root of a binary tree, not a BST). You may add helper functions or additional parameters if you wish. Your code must be recursive.

- a) `int sum_to_depth(BTNode *t, int n)` – return the sum of all nodes in t with depth $\leq n$ (assume the root is at depth 0)
- b) `int biggest_internal(BTNode *t)` – return the largest value in t that is not stored in a leaf node
- c) `void set_parity(BTNode *t)` – for all nodes in t : if the node is a left child set its value to -1, if the node is a right child set its value to 1, and if the node has no parents, set its value to 0

```
typedef struct BinaryTreeNode{
    int data;
    struct BinaryTreeNode* left;
    struct BinaryTreeNode* right;
}BTNode;
```


Question 3 [3 marks]

Answer the following questions in 2 sentences or less per question:

As discussed in Unit 6:

- a) What are 2 benefits of modularity?
- b) What are 2 properties of a good API?
- c) What are 2 things that can be done in C++ that cannot be done in C?

