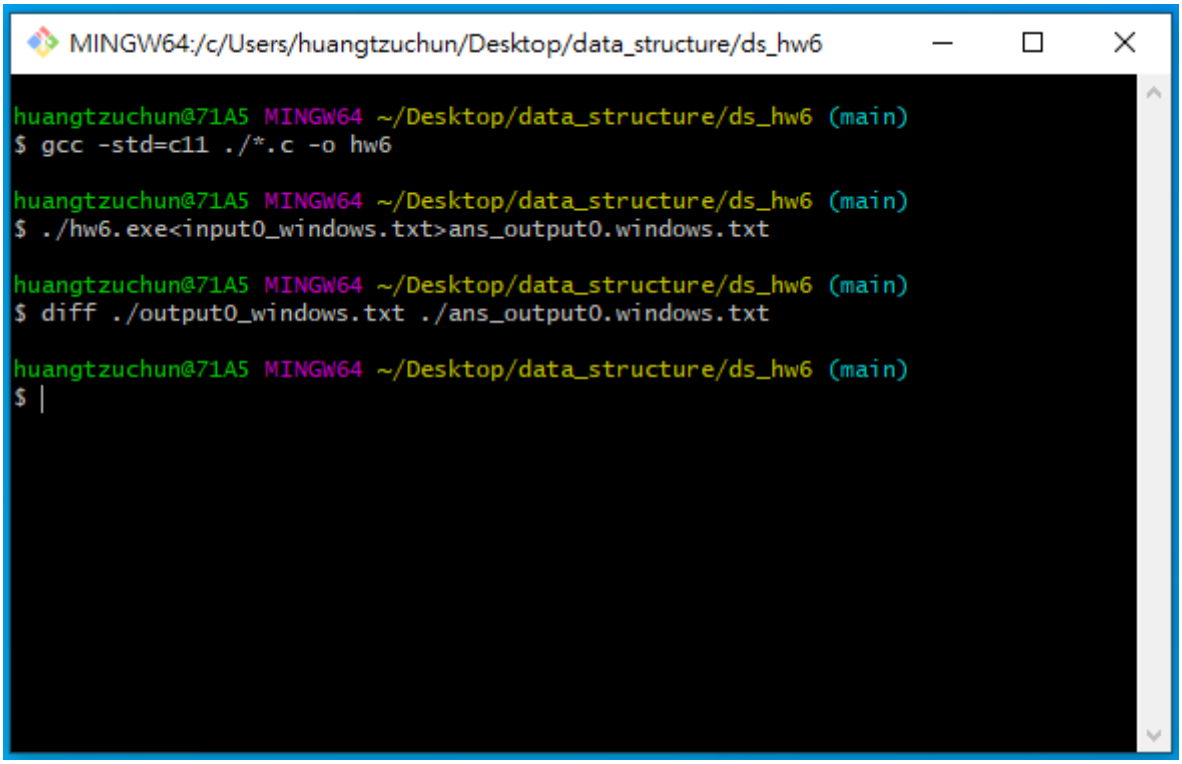
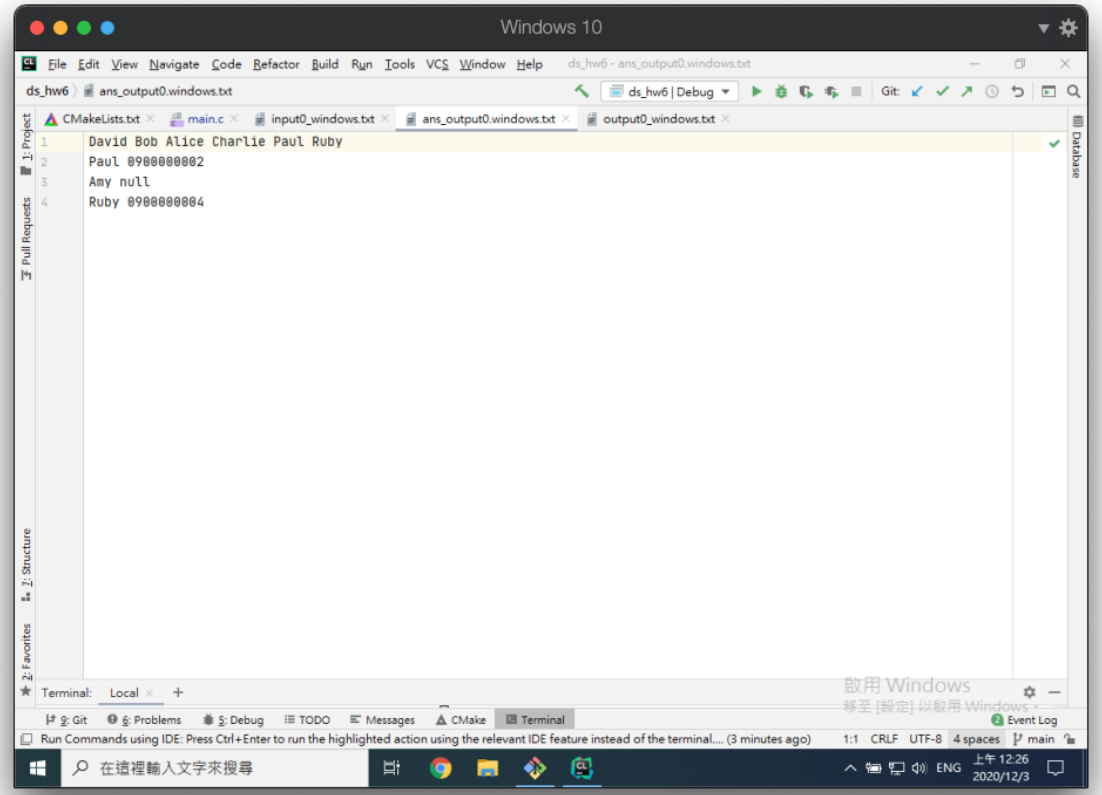
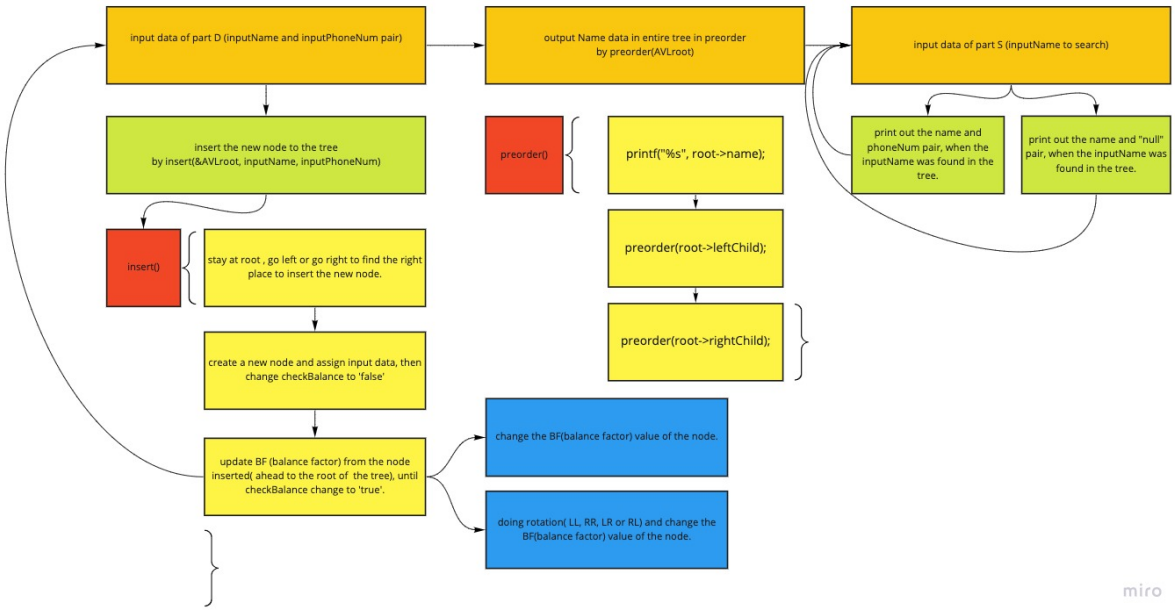
E14084117_黃子峻

(1) result screenshot





(2) program structure



(3) program functions

`void LLrotation(treePointer* root)`

> Function for doing LL rotation.

parameters

- `treePointer* root` : the tree node to start LL rotation.(which has BF = 2)

return values

- No return value (void).

`void RRrotation(treePointer* root)`

> Function for doing RR rotation.

parameters

- `treePointer* root` : the tree node to start RR rotation.(which has BF = -2)

return values

- No return value (void).

`void LRrotation(treePointer* root)`

> Function for doing LR rotation.

parameters

- `treePointer* root` : the tree node to start LR rotation.(which has BF = 2)

return values

- No return value (void).

`void RLrotation(treePointer* root)`

> Function for doing RL rotation.

parameters

- `treePointer* root` : the tree node to start RL rotation.(which has BF = -2)

return values

- No return value (void).

`void insert(treePointer root, char addName, char* addNum)`

> Function for insert a new node to the tree with addName and addNum data.

parameters

- `treePointer* root` : the tree node to start finding the right place for insert the new node.

return values

- No return value (void).

`void preorder(treePointer root)`

> print out the preorder traversal result.

parameters

- `treePointer root` : the tree node to start preorder traversal. By using preorder(AVLroot) to traversal entire tree by preorder.

return values

- No return value (void).

`bool search(treePointer root,char* find_name)`

> Function for finding node with find_name in the tree and print out it's name and phoneNum pair.(print out name and "null" pair when finding failure.)

parameters

- `treePointer root` : the tree node to start finding. By using `search(AVLroot, find_name)` to search the name in entire tree.
- `char* find_name` : the name we want to search in the tree, which has length equal or less to 20 characters.

return values

- `bool true` : found the node with `find_name` in the tree.
- `boo false` : not found the node with `find_name` in the tree.

(4)self define structure

```
typedef struct treeNode *treePointer;

struct treeNode {
    treePointer leftChild;
    treePointer rightChild;
    int BF;
    char phoneNum[11];
    char name[21];
};
```

- `leftChild/rightChild` : treePointer point to the child of this node.
- `int BF` : balance factor of this node, can be -1, 0 or 1.
- `char phoneNum[11]` : phone number data in char array.
- `char name[21]` : phone number data in char array.

(5)global variables

`treePointer AVLroot`

- The tree pointer point to the root of AVL tree.
- It point to NULL at beginning.

`bool checkBalance`

- It's set to true at beginning.

- `false` for label still needing to check balance.
- `true` for stop checking balance in the current turn.