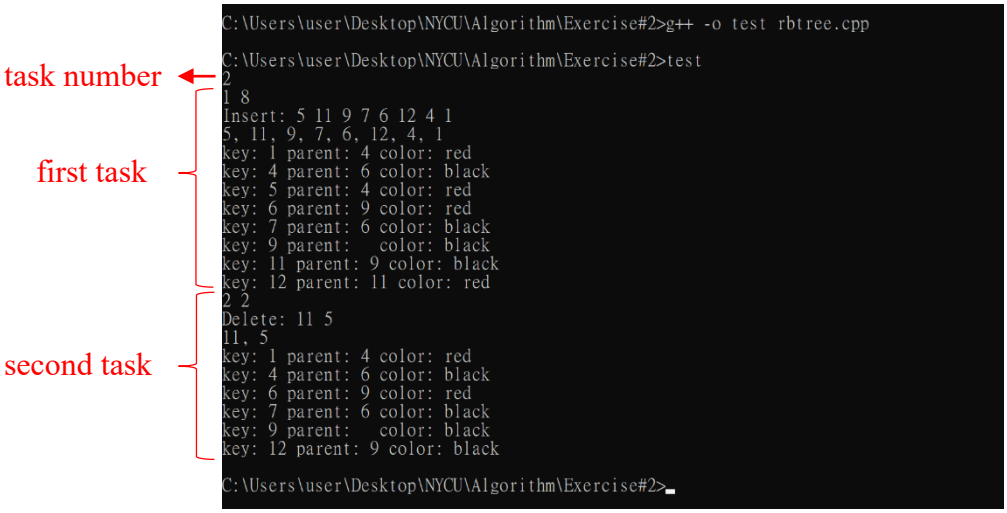
Ex2 - RBtree

- **Environment**

- Windows
- g++
- Dev C++ 5.11
- How to run my program: (in CMD)



- **Solutions**

StepI: define 2 classes

- TreeNode: the basic unit of a tree, including key, color, parent, lchild and rchild
- RBTree: the designated data structure, represented by its root node

StepII: create some functions (as tools)

- Transplant: replace one node with the other (referred to the ppt of Lec13)
- LeftRotate: rotate the tree counterclockwise (referred to the ppt of Lec13)
- RightRotate: rotate the tree clockwise (similar to LeftRotate)
- Search: search the node whose key equals the required value, used when deleting a TreeNode
- Min: look for the minimal node rooted at a specified node, used when looking for the successor of a TreeNode (referred to the ppt of Lec12)

StepIII: create more functions (for the main purpose)

- Insert: insert a new node based on the BinaryTree properties (referred to the ppt of Lec13)
- InsertFixup: rebuild the tree in order to fulfill the RBTree properties (referred to the ppt of Lec13)

- Delete: delete a node based on the BinaryTree properties (referred to the ppt of Lec13)
- DeleteFixup: rebuild the tree in order to fulfill the RBTree properties (referred to the ppt of Lec13)

StepIV: create additional functions (for display & termination)

- Inorder: display the RBTree by inorder traversal, composed of a driver and a workhorse
- DeleteTree: retrieve the space occupied by the remaining TreeNodes at the end of the program by postorder traversal, composed of a driver and a workhorse

StepV: write the main function

- Input the number of tasks t → run the for loop t times

(in each for loop:)

- Input op & n → if op is 1, do the insertion, otherwise do the deletion n times
- Input the elements → get the key value from standard input. if op is 1, create a new TreeNode, otherwise call the Search function to find the TreeNode to be deleted
- Display the RBTree → call the Inorder driver
- Termination → call the DeleteTree driver to retrieve the occupied space