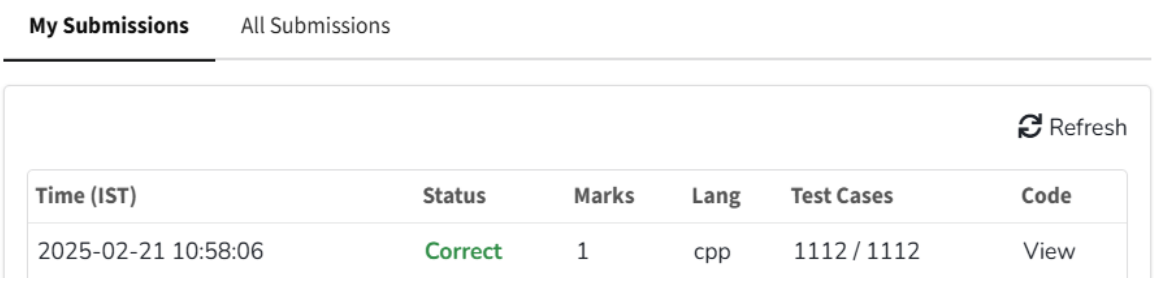
ASSIGNMENT

Student Name: Sandhya Bharti **UID: 22BCS11412**

Branch: BE-CSE **Section/Group: 608/B**

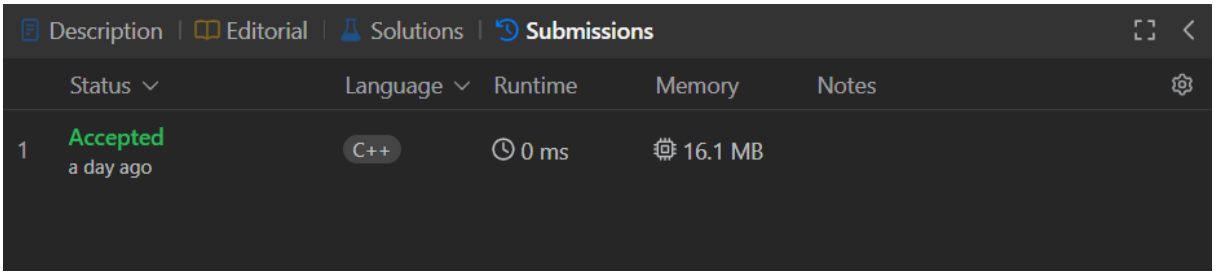
Semester: 6th **Subject Name: AP LAB**

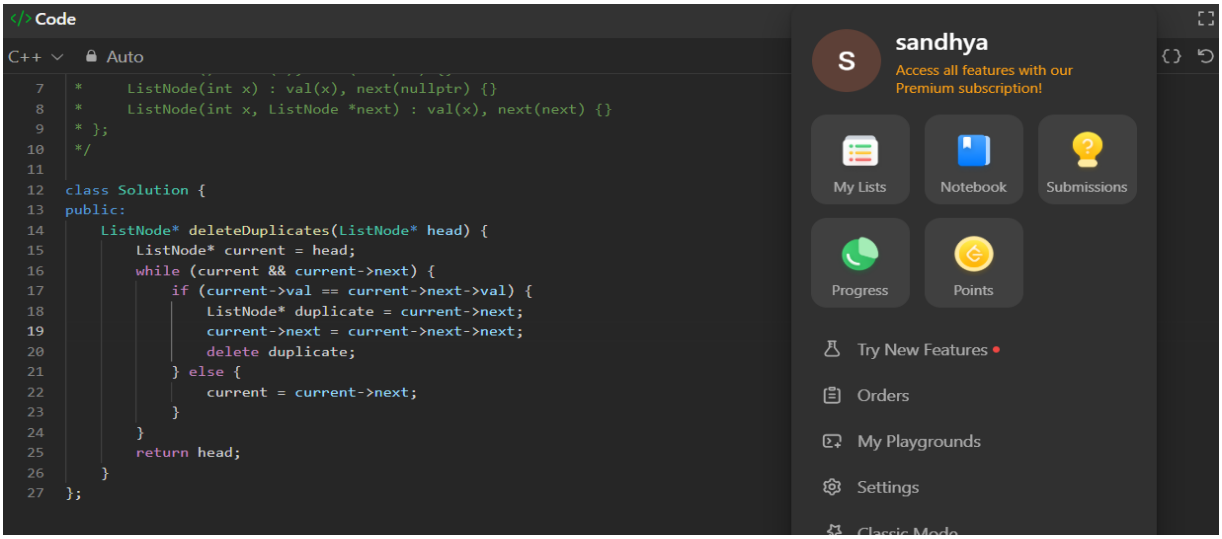
1. **Print Linked List**



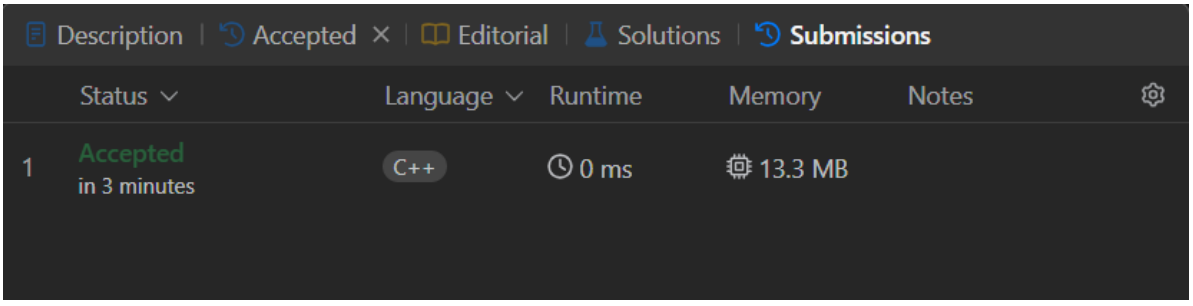

```
class Solution {
  public:
    // Function to display the elements of a linked list in the same line
    void printList(Node *head) {
        Node* temp = head;
        while (temp) {
            cout << temp->data;
            if(temp->next) cout<< " ";// Print data followed by space
            temp = temp->next;
        }
     // Newline at the end
    }
};
// } Driver Code Ends
```

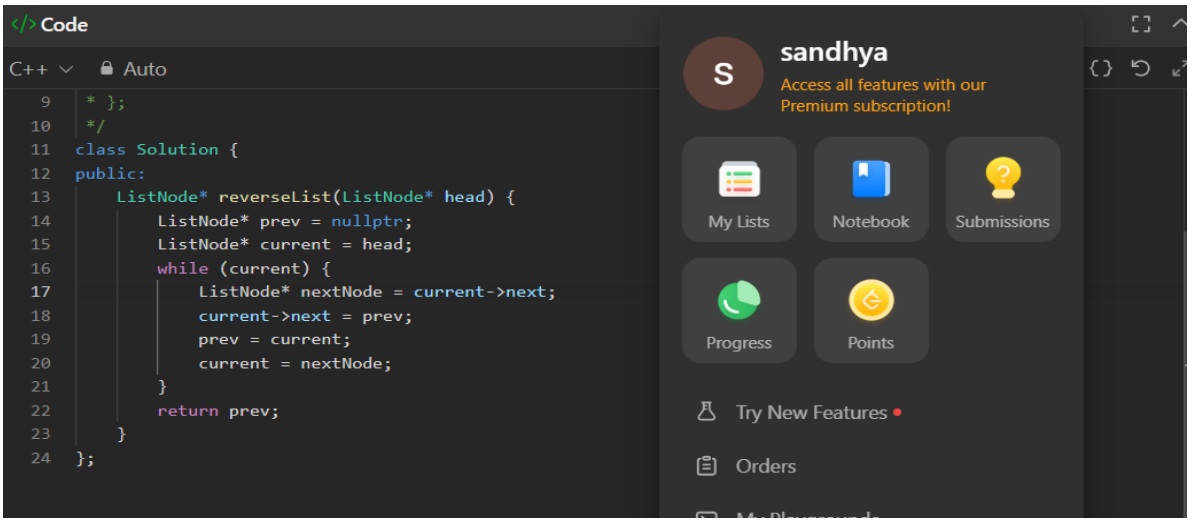
2. Remove duplicates from a sorted list:



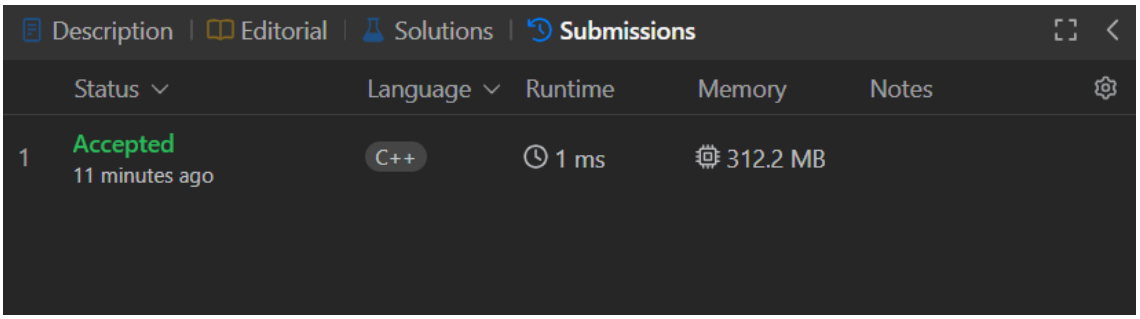


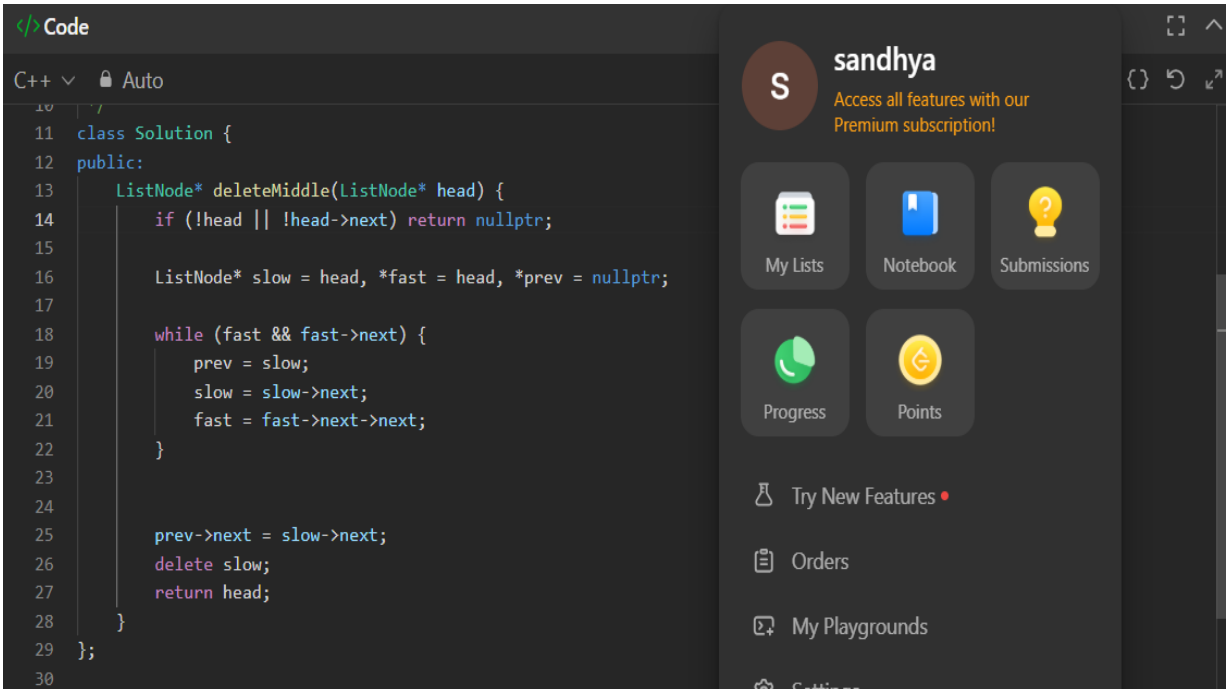
3. Reverse a linked list:



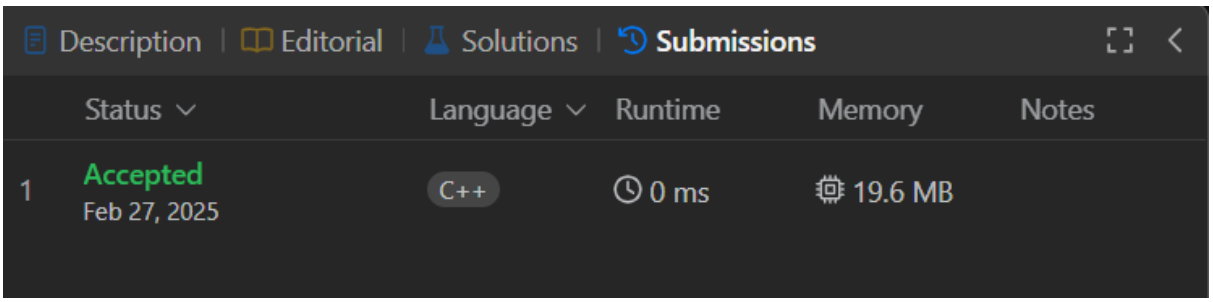


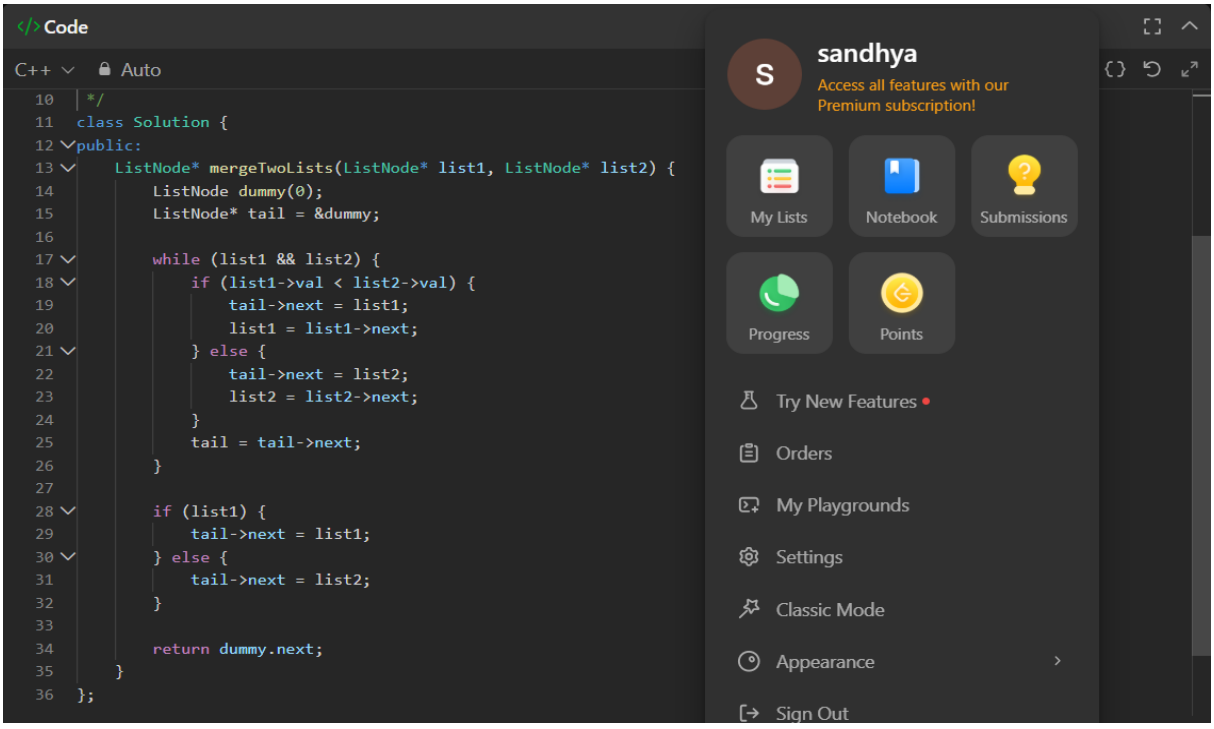
4. Delete middle node of a list:



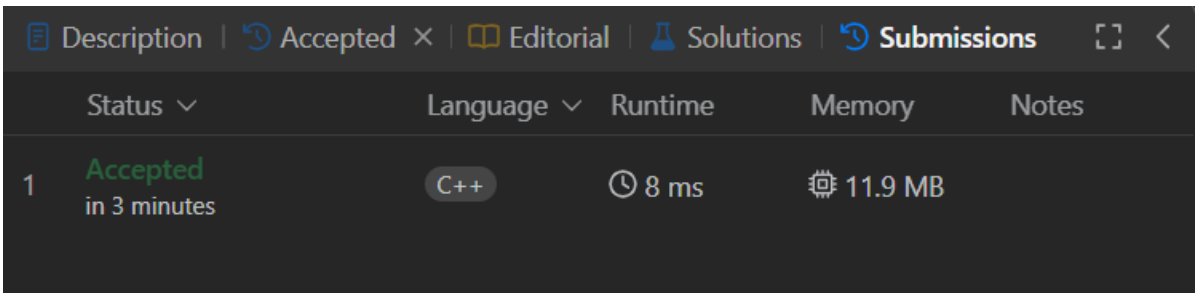


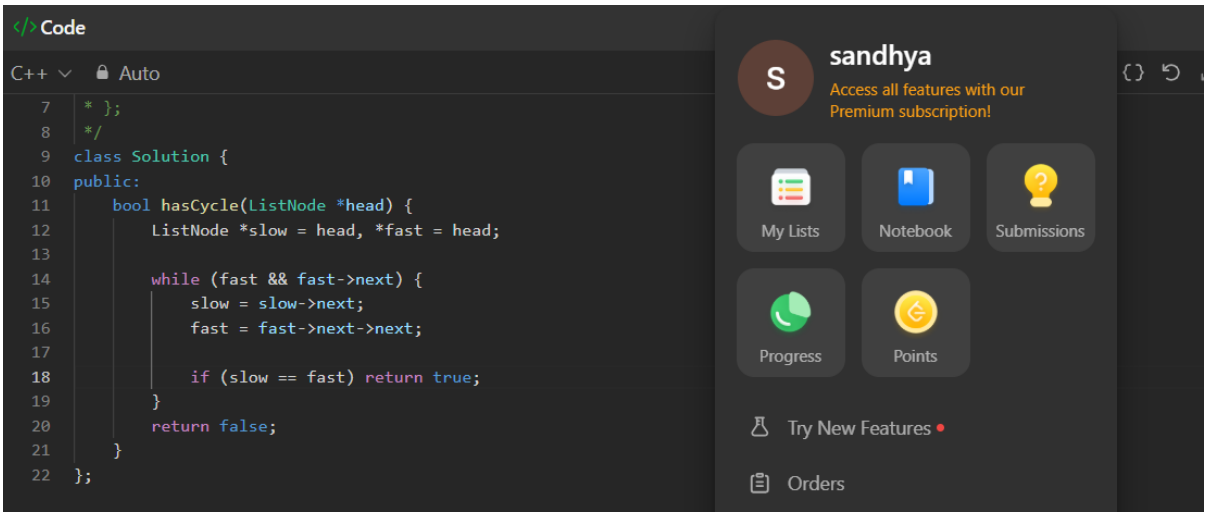
5. Merge two sorted linked lists:



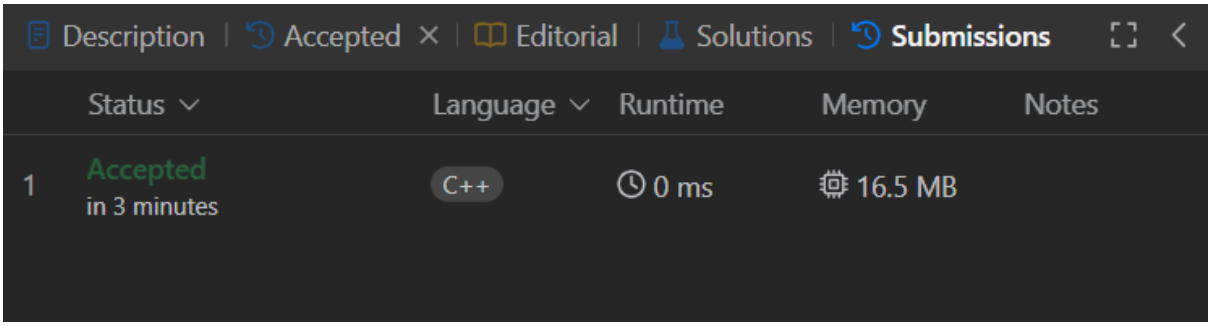


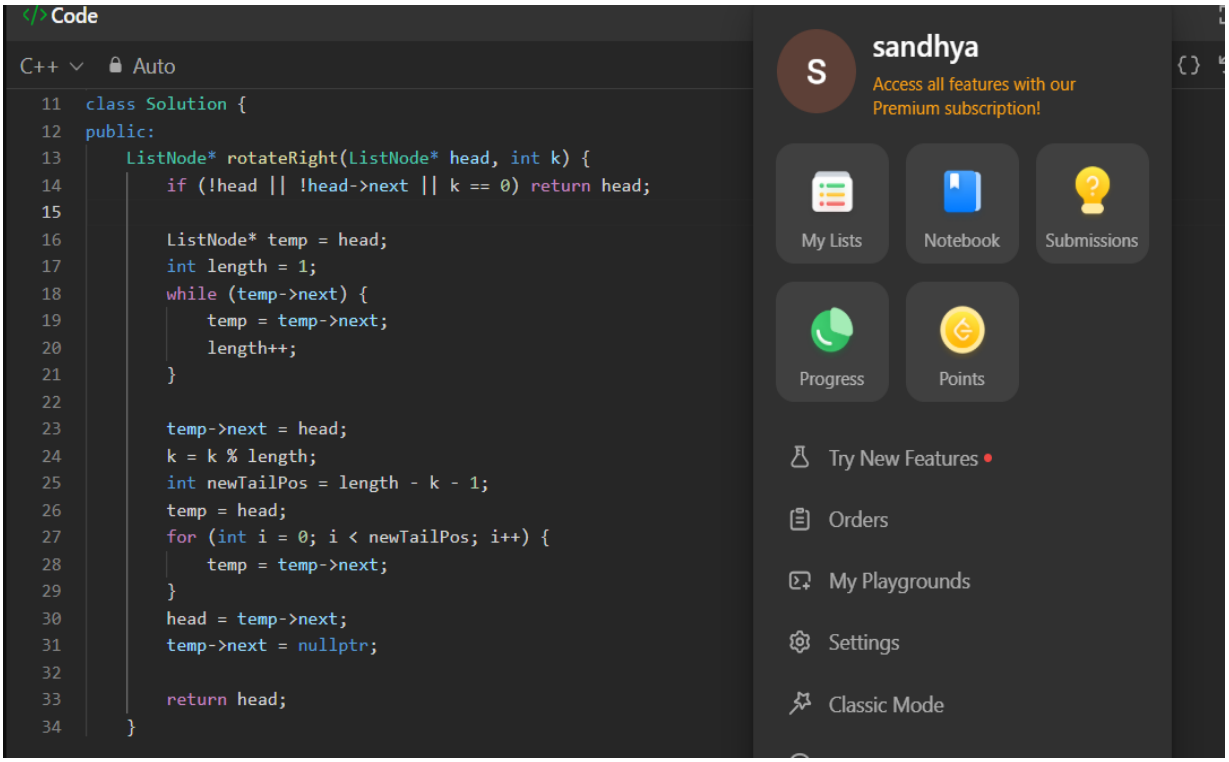
6. Detect a cycle in a linked list:



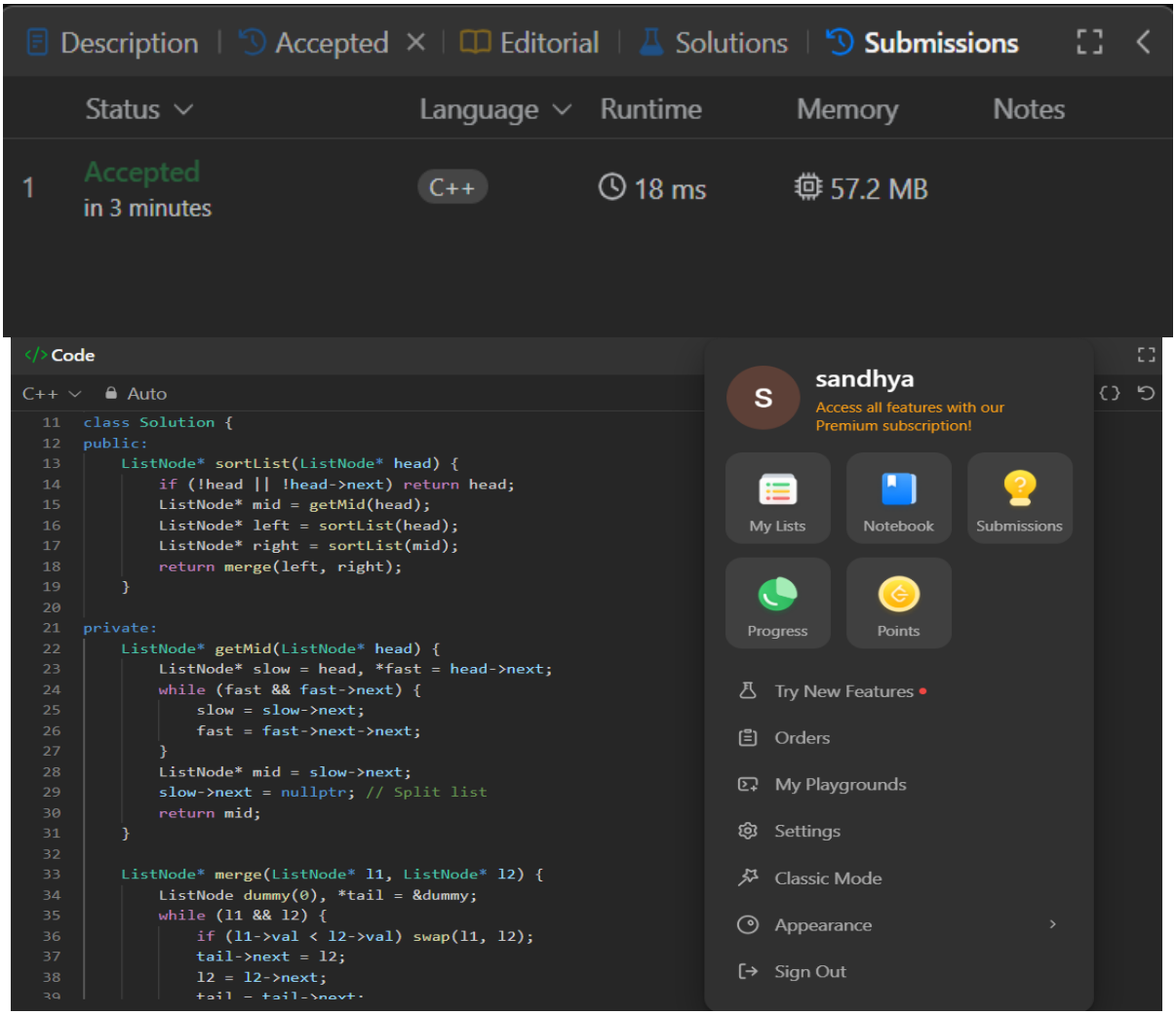


7. Rotate a list:

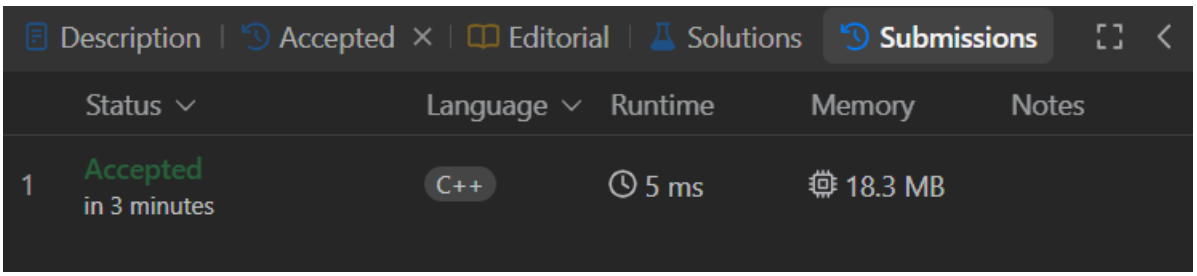


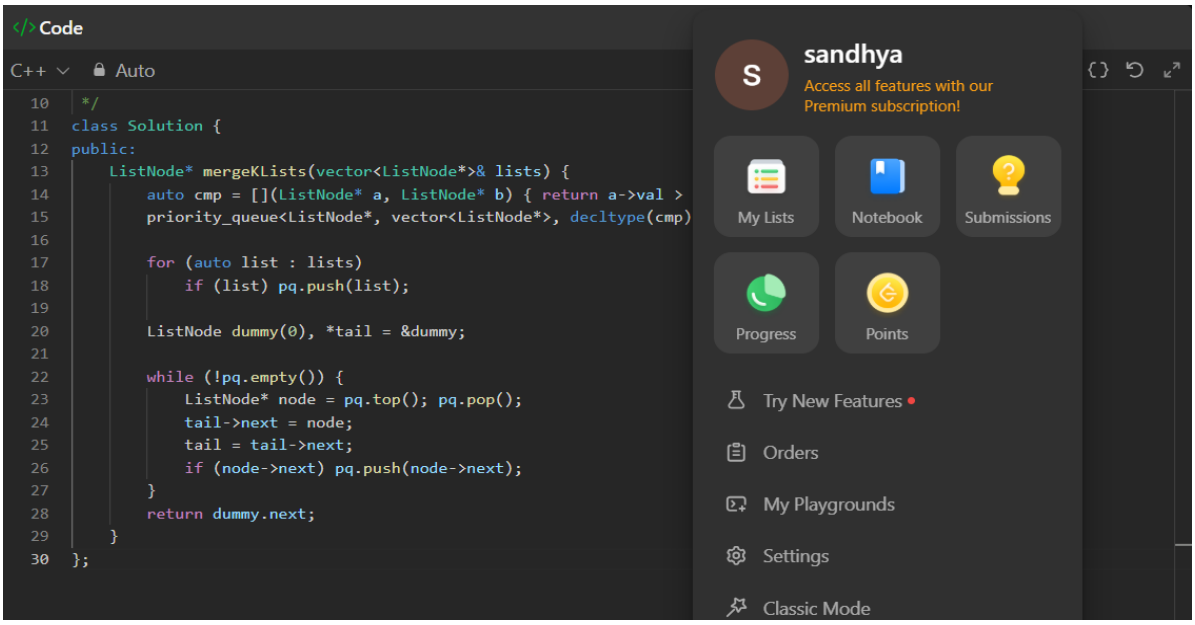


8. Sort List:



9. Merge k sorted lists:




Code
C++
Auto
*/
class Solution {
public:
ListNode* mergeKLists(vector<ListNode*>& lists) {
auto cmp = [](ListNode* a, ListNode* b) { return a->val >
priority_queue<ListNode*, vector<ListNode*>, decltype(cmp)
for (auto list : lists)
if (list) pq.push(list);
ListNode dummy(0), *tail = &dummy;
while (!pq.empty()) {
ListNode* node = pq.top(); pq.pop();
tail->next = node;
tail = tail->next;
if (node->next) pq.push(node->next);
}
return dummy.next;
}
};
S
sandhya
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode