|  |  |
| --- | --- |
| **WORKSHEET-1.1** | |
| **Student Name: Shubham Kumar** | **UID: 21BCS7104** |
| **Branch: CSE** | **Section/Group: 614-A** |
| **Semester: 6th** | **Date of Performance: 13-01-2024** |
| **Subject Name: Advanced Programming Lab-2** | **Subject Code: 21CSP-351** |

**1. Aim:** Arrays, Stacks, Queues linked list

A) Return the minimum number of jumps to reach nums[n - 1]. The test cases are generated such that you can reach nums[n - 1].

B) You are given the heads of two sorted linked lists list1 and list2.Merge the two lists into one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list.

**2. Source Code/Output:**

**A)**

class Solution {

public:

int jump(vector<int>& nums) {

int ans = 0;

int end = 0;

int far = 0;

for (int i = 0; i < nums.size() - 1; ++i) {

far = max(far, i + nums[i]);

if (far >= nums.size() - 1) {

++ans;

break;

}

if (i == end) {

++ans;

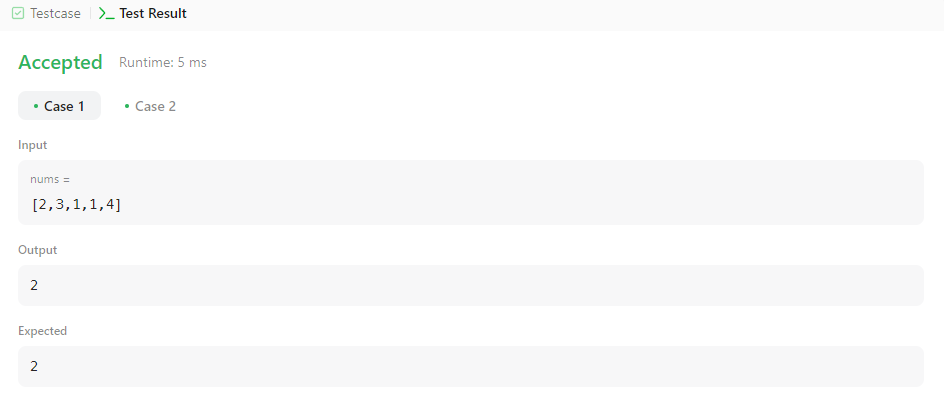
end = far;

}

}

return ans;

}

};****

B)

class Solution {

public:

ListNode\* mergeTwoLists(ListNode\* list1, ListNode\* list2) {

if (!list1 || !list2)

return list1 ? list1 : list2;

if (list1->val > list2->val)

swap(list1, list2);

list1->next = mergeTwoLists(list1->next, list2);

return list1;

}

};

