

## DAY 2

Maadhav Hira

22BCS10380

KPIT-901(A)

### Q : Majority Elements

```
day2ques1.cpp > reverseList(ListNode *)
1 //question number 1
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 int majorityElement(vector<int>& nums) {
7     int candidate = 0, count = 0;
8     for (int num : nums) {
9         if (count == 0) {
10             candidate = num;
11         }
12         count += (num == candidate) ? 1 : -1;
13     }
14     count = 0;
15     for (int num : nums) {
16         if (num == candidate) {
17             count++;
18         }
19     }
20     if (count > nums.size() / 2) {
21         return candidate;
22     }
23     return -1;
24 }
25
26 int main() {
27     int n;
28     cout << "Enter the size of the array: ";
29     cin >> n;
30     vector<int> nums(n);
31     cout << "Enter " << n << " elements of the array: ";
32     for (int i = 0; i < n; ++i) {
33         cin >> nums[i];
34     }
35     cout << "Majority Element: " << majorityElement(nums) << endl;
36     return 0;
37 }
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\tempCodeRunnerFile
Enter the size of the array: 5
Enter 5 elements of the array: 2 2 2 3 3
Majority Element: 2
```

## Question:. Pascal's Triangle

```
1 // question 2
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 vector<vector<int>> generatePascalsTriangle(int numRows) {
7     vector<vector<int>> triangle;
8
9     for (int i = 0; i < numRows; ++i) {
10         vector<int> row(i + 1, 1);
11         for (int j = 1; j < i; ++j) {
12             row[j] = triangle[i - 1][j - 1] + triangle[i - 1][j];
13         }
14         triangle.push_back(row);
15     }
16     return triangle;
17 }
18
19 int main() {
20     int numRows;
21     cout << "Enter the number of rows for Pascal's Triangle: ";
22     cin >> numRows;
23     vector<vector<int>> pascalsTriangle = generatePascalsTriangle(numRows);
24     cout << "Pascal's Triangle:" << endl;
25     for (const auto& row : pascalsTriangle) {
26         for (int num : row) {
27             cout << num << " ";
28         }
29         cout << endl;
30     }
31     return 0;
32 }
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd C:\Users\Gaurav Kumar\OneDrive\Desktop\class\ && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && C:\Users\Gaurav Kumar\OneDrive\Desktop\class\tempCodeRunnerFile
Enter the number of rows for Pascal's Triangle: 5
Pascal's Triangle:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

## Question: Container With Most Water

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int maxArea(vector<int>& height) {
    int left = 0, right = height.size() - 1;
    int max_water = 0;

    while (left < right) {
        int width = right - left;
        int current_height = min(height[left], height[right]);
        max_water = max(max_water, width * current_height);

        if (height[left] < height[right]) {
            ++left;
        } else {
            --right;
        }
    }

    return max_water;
}

int main() {
    int n;
    cout << "Enter the number of lines: ";
    cin >> n;

    vector<int> height(n);
    cout << "Enter the heights of the lines: ";
    for (int i = 0; i < n; ++i) {
        cin >> height[i];
    }

    cout << "Maximum Water: " << maxArea(height) << endl;
    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\tempCodeRunnerFile
Enter the number of lines: 9
Enter the heights of the lines: 1 8 6 2 5 4 8 3 7
Maximum Water: 49
```

## Question. Merge Two Sorted Lists

```
115 //question 4
116 #include <iostream>
117 #include <vector>
118 using namespace std;
119
120 vector<int> mergeSortedArrays(const vector<int>& arr1, const vector<int>& arr2) {
121     vector<int> merged;
122     int i = 0, j = 0;
123     while (i < arr1.size() && j < arr2.size()) {
124         if (arr1[i] <= arr2[j]) {
125             merged.push_back(arr1[i]);
126             i++;
127         } else {
128             merged.push_back(arr2[j]);
129             j++;
130         }
131     }
132
133     while (i < arr1.size()) {
134         merged.push_back(arr1[i]);
135         i++;
136     }
137
138     while (j < arr2.size()) {
139         merged.push_back(arr2[j]);
140         j++;
141     }
142
143     return merged;
144 }
145
146 int main() {
147     int n1, n2;
148     cout << "Enter the size of the first sorted array: ";
149     cin >> n1;
150
151     vector<int> arr1(n1);
152     cout << "Enter the elements of the first sorted array: ";
153     for (int i = 0; i < n1; i++) {
154         cin >> arr1[i];
155     }
156 }
```

```
156
157     cout << "Enter the size of the second sorted array: ";
158     cin >> n2;
159
160     vector<int> arr2(n2);
161     cout << "Enter the elements of the second sorted array: ";
162     for (int i = 0; i < n2; i++) {
163         cin >> arr2[i];
164     }
165
166     vector<int> result = mergeSortedArrays(arr1, arr2);
167
168     cout << "Merged Sorted Array: ";
169     for (int val : result) {
170         cout << val << " ";
171     }
172     cout << endl;
173
174     return 0;
175 }
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "C:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "C:\Users\Gaurav Kumar\OneDrive\Desktop\class\tempCodeRunnerFile
Enter the size of the first sorted array: 5
Enter the elements of the first sorted array: 1 2 3 4 5
Enter the size of the second sorted array: 5
Enter the elements of the second sorted array: 1 2 5 7 8
Merged Sorted Array: 1 1 2 2 3 4 5 5 7 8
```

## Question. Reverse Linked List

```
178 //question 5
179 #include <iostream>
180 using namespace std;
181
182 struct ListNode {
183     int val;
184     ListNode* next;
185     ListNode(int x) : val(x), next(nullptr) {}
186 };
187
188 ListNode* reverselist(ListNode* head) {
189     ListNode* prev = nullptr;
190     ListNode* current = head;
191
192     while (current != nullptr) {
193         ListNode* nextNode = current->next;
194         current->next = prev;
195         prev = current;
196         current = nextNode;
197     }
198
199     return prev;
200 }
201
202 void printList(ListNode* head) {
203     while (head != nullptr) {
204         cout << head->val << " ";
205         head = head->next;
206     }
207     cout << endl;
208 }
209
210 ListNode* createlist(int n) {
211     if (n == 0) return nullptr;
212     int val;
213     cout << "Enter the value of node 1: ";
214     cin >> val;
215     ListNode* head = new ListNode(val);
216     ListNode* current = head;
217     for (int i = 2; i <= n; ++i) {
218         cout << "Enter the value of node " << i << ": ";
219         cin >> val;
220         current->next = new ListNode(val);
221         current = current->next;
222     }
223     return head;
224 }
225
226 int main() {
227     int n;
228     cout << "Enter the number of nodes in the linked list: ";
229     cin >> n;
230
231     ListNode* head = createlist(n);
232
233     cout << "Original Linked List: ";
234     printList(head);
235
236     head = reverselist(head);
237
238     cout << "Reversed Linked List: ";
239     printList(head);
240
241     return 0;
242 }
```

```
219         current->next = new ListNode(val);
220         current = current->next;
221     }
222     return head;
223 }
224
225 int main() {
226     int n;
227     cout << "Enter the number of nodes in the linked list: ";
228     cin >> n;
229
230     ListNode* head = createlist(n);
231
232     cout << "Original Linked List: ";
233     printList(head);
234
235     head = reverselist(head);
236
237     cout << "Reversed Linked List: ";
238     printList(head);
239
240     return 0;
241 }
242
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "c:\Use
rs\Gaurav Kumar\OneDrive\Desktop\class\tempCodeRunnerFile
```

Enter the number of nodes in the linked list: 5

Enter the value of node 1: 5

Enter the value of node 2: 6

Enter the value of node 3: 5

Enter the value of node 4: 4

Enter the value of node 5: 7

Original Linked List: 5 6 5 4 7

Reversed Linked List: 7 4 5 6 5