DAY 2

Gaurav Kumar **22BCS10159** **KPIT-901(A)**

Q : Majority Elements

day2ques1.cpp > reverseList(ListNode *)

```cpp
//question number 1
#include <iostream>
#include <vector>
using namespace std;

int majorityElement(vector<int>& nums) {
    int candidate = 0, count = 0;
    for (int num : nums) {
        if (count == 0) {
            candidate = num;
        }
        count += (num == candidate) ? 1 : -1;
    }
    count = 0;
    for (int num : nums) {
        if (num == candidate) {
            count++;
        }
    }
    if (count > nums.size() / 2) {
        return candidate;
    }
    return -1;
}

int main() {
    int n;
    cout << "Enter the size of the array: ";
    cin >> n;
    vector<int> nums(n);
    cout << "Enter " << n << " elements of the array: ";
    for (int i = 0; i < n; ++i) {
        cin >> nums[i];
    }
    cout << "Majority Element: " << majorityElement(nums) << 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 size of the array: 5
Enter 5 elements of the array: 2 2 2 3 3
Majority Element: 2
```

Question:. Pascal's Triangle

```cpp
// question 2
#include <iostream>
#include <vector>
using namespace std;

vector<vector<int>> generatePascalsTriangle(int numRows) {
    vector<vector<int>> triangle;

    for (int i = 0; i < numRows; ++i) {
        vector<int> row(i + 1, 1);
        for (int j = 1; j < i; ++j) {
            row[j] = triangle[i - 1][j - 1] + triangle[i - 1][j];
        }
        triangle.push_back(row);
    }
    return triangle;
}

int main() {
    int numRows;
    cout << "Enter the number of rows for Pascal's Triangle: ";
    cin >> numRows;
    vector<vector<int>> pascalsTriangle = generatePascalsTriangle(numRows);
    cout << "Pascal's Triangle:" << endl;
    for (const auto& row : pascalsTriangle) {
        for (int num : row) {
            cout << num << " ";
        }
        cout << 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 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

```cpp
#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

```cpp
//question 4
#include <iostream>
#include <vector>
using namespace std;

vector<int> mergeSortedArrays(const vector<int>& arr1, const vector<int>& arr2) {
    vector<int> merged;
    int i = 0, j = 0;
    while (i < arr1.size() && j < arr2.size()) {
        if (arr1[i] <= arr2[j]) {
            merged.push_back(arr1[i]);
            i++;
        } else {
            merged.push_back(arr2[j]);
            j++;
        }
    }

    while (i < arr1.size()) {
        merged.push_back(arr1[i]);
        i++;
    }

    while (j < arr2.size()) {
        merged.push_back(arr2[j]);
        j++;
    }

    return merged;
}

int main() {
    int n1, n2;
    cout << "Enter the size of the first sorted array: ";
    cin >> n1;

    vector<int> arr1(n1);
    cout << "Enter the elements of the first sorted array: ";
    for (int i = 0; i < n1; i++) {
        cin >> arr1[i];
    }
```

```cpp
    cout << "Enter the size of the second sorted array: ";
    cin >> n2;

    vector<int> arr2(n2);
    cout << "Enter the elements of the second sorted array: ";
    for (int i = 0; i < n2; i++) {
        cin >> arr2[i];
    }

    vector<int> result = mergeSortedArrays(arr1, arr2);

    cout << "Merged Sorted Array: ";
    for (int val : result) {
        cout << val << " ";
    }
    cout << endl;

    return 0;
}
```

```
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 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

```cpp
//question 5
#include <iostream>
using namespace std;

struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x), next(nullptr) {}
};

ListNode* reverseList(ListNode* head) {
    ListNode* prev = nullptr;
    ListNode* current = head;

    while (current != nullptr) {
        ListNode* nextNode = current->next;
        current->next = prev;
        prev = current;
        current = nextNode;
    }

    return prev;
}

void printList(ListNode* head) {
    while (head != nullptr) {
        cout << head->val << " ";
        head = head->next;
    }
    cout << endl;
}
ListNode* createList(int n) {
    if (n == 0) return nullptr;
    int val;
    cout << "Enter the value of node 1: ";
    cin >> val;
    ListNode* head = new ListNode(val);
    ListNode* current = head;
    for (int i = 2; i <= n; ++i) {
        cout << "Enter the value of node " << i << ": ";
        cin >> val;
        current->next = new ListNode(val);
```

```cpp
        current->next = new ListNode(val);
        current = current->next;
    }
    return head;
}

int main() {
    int n;
    cout << "Enter the number of nodes in the linked list: ";
    cin >> n;

    ListNode* head = createList(n);

    cout << "Original Linked List: ";
    printList(head);

    head = reverseList(head);

    cout << "Reversed Linked List: ";
    printList(head);

    return 0;
}

```

```
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
```