1. Write a program to accept n and store the elements into the array of size n.

#include <iostream>

using namespace std;

int main() {

    int n;

    cout << "Enter the size of the array: ";

    cin >> n;

    int arr[n];  // Create array of size n

    cout << "Enter " << n << " elements:" << endl;

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    // Display the entered array

    cout << "You entered the array: ";

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    return 0;

}

2. Reverse the given array.

#include <iostream>

using namespace std;

int main() {

    int n;

    cout << "Enter the size of the array: ";

    cin >> n;

    int arr[n];

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    // Display original array

    cout << "Original array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    // Reverse array in-place

    for (int i = 0; i < n / 2; i++) {

        int temp = arr[i];

        arr[i] = arr[n - 1 - i];

        arr[n - 1 - i] = temp;

    }

    // Display reversed array

    cout << "Reversed array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    return 0;

}

3. Sort the array in ascending or descending order based on input of user.

#include <iostream>

#include <algorithm>  // for sort()

using namespace std;

int main() {

    int n;

    cout << "Enter the size of the array: ";

    cin >> n;

    int arr[n];

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    char order;

    cout << "Enter sorting order (A for Ascending, D for Descending): ";

    cin >> order;

    if (order == 'A' || order == 'a') {

        sort(arr, arr + n);  // Ascending sort

    } else if (order == 'D' || order == 'd') {

        sort(arr, arr + n, greater<int>());  // Descending sort

    } else {

        cout << "Invalid input! Defaulting to ascending order.\n";

        sort(arr, arr + n);

    }

    cout << "Sorted array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    return 0;

}

4. Implement Binary Search on the array.

#include <iostream>

#include <algorithm>  // for sort()

using namespace std;

int main() {

    int n;

    cout << "Enter the size of the array: ";

    cin >> n;

    int arr[n];

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    // Binary search requires sorted array

    sort(arr, arr + n);

    cout << "Sorted array for binary search: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    int key;

    cout << "Enter the element to search: ";

    cin >> key;

    // Binary Search

    int low = 0, high = n - 1;

    bool found = false;

    while (low <= high) {

        int mid = (low + high) / 2;

        if (arr[mid] == key) {

            cout << "Element found at index (0-based): " << mid << endl;

            found = true;

            break;

        } else if (arr[mid] < key) {

            low = mid + 1;

        } else {

            high = mid - 1;

        }

    }

    if (!found)

        cout << "Element not found in the array." << endl;

    return 0;

}

5. Write a program to create a 2D array and display its elements row-wise.

#include <iostream>

using namespace std;

int main() {

    int rows, cols;

    cout << "Enter number of rows: ";

    cin >> rows;

    cout << "Enter number of columns: ";

    cin >> cols;

    int arr[rows][cols];

    // Input elements

    cout << "Enter elements for 2D array:\n";

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            cin >> arr[i][j];

        }

    }

    // Display row-wise

    cout << "2D array displayed row-wise:\n";

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            cout << arr[i][j] << " ";

        }

        cout << endl;  // Move to next row

    }

    return 0;

}

6. Create a program to compute the sum of all elements in a 2D array.

#include <iostream>

using namespace std;

int main() {

    int rows, cols;

    cout << "Enter number of rows: ";

    cin >> rows;

    cout << "Enter number of columns: ";

    cin >> cols;

    int arr[rows][cols];

    // Input elements

    cout << "Enter elements for 2D array:\n";

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            cin >> arr[i][j];

        }

    }

    // Compute sum

    int sum = 0;

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            sum += arr[i][j];

        }

    }

    cout << "Sum of all elements in the 2D array: " << sum << endl;

    return 0;

}

7. Write a program to check if a given element exists in a 2D array.

#include <iostream>

using namespace std;

int main() {

    int rows, cols;

    cout << "Enter number of rows: ";

    cin >> rows;

    cout << "Enter number of columns: ";

    cin >> cols;

    int arr[rows][cols];

    // Input elements

    cout << "Enter elements for 2D array:\n";

    for (int i = 0; i < rows; i++)

        for (int j = 0; j < cols; j++)

            cin >> arr[i][j];

    int key;

    cout << "Enter the element to search: ";

    cin >> key;

    // Search for element

    bool found = false;

    for (int i = 0; i < rows && !found; i++) {

        for (int j = 0; j < cols; j++) {

            if (arr[i][j] == key) {

                cout << "Element found at position (row, col): (" << i << ", " << j << ")\n";

                found = true;

                break;

            }

        }

    }

    if (!found)

        cout << "Element not found in the 2D array.\n";

    return 0;

}

Prithika S

RA2211003010045