#include <iostream>

using namespace std;

int main() {

    int arr[100], size = 0, choice;

    do {

        cout << "\n1. Create\n2. Display\n3. Insert\n4. Delete\n5. Search\n6. Exit\n";

        cout << "Choice: ";

        cin >> choice;

        if (choice == 1) {

            cout << "Enter size: ";

            cin >> size;

            cout << "Enter elements: ";

            for (int i = 0; i < size; i++)

                cin >> arr[i];

        }

        else if (choice == 2) {

            for (int i = 0; i < size; i++)

                cout << arr[i] << " ";

            cout << endl;

        }

        else if (choice == 3) {

            int pos, val;

            cout << "Position: ";

            cin >> pos;

            cout << "Value: ";

            cin >> val;

            for (int i = size; i > pos; i--)

                arr[i] = arr[i - 1];

            arr[pos] = val;

            size++;

        }

        else if (choice == 4) {

            int pos;

            cout << "Position: ";

            cin >> pos;

            for (int i = pos; i < size - 1; i++)

                arr[i] = arr[i + 1];

            size--;

        }

        else if (choice == 5) {

            int val;

            cout << "Search for: ";

            cin >> val;

            for (int i = 0; i < size; i++) {

                if (arr[i] == val) {

                    cout << "Found at index " << i << endl;

                    break;

                }

                if (i == size - 1)

                    cout << "Not found\n";

            }

        }

    } while (choice != 6);

    return 0;

}

Q2) #include <iostream>

#include <vector>

#include <set>

using namespace std;

int main() {

    int n;

    cout << "Enter size: ";

    cin >> n;

    vector<int> arr(n);

    cout << "Enter elements: ";

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    set<int> unique\_elements\_set(arr.begin(), arr.end());

    vector<int> unique\_arr(unique\_elements\_set.begin(), unique\_elements\_set.end());

    cout << "Unique elements: ";

    for (size\_t i = 0; i < unique\_arr.size(); i++) {

        cout << unique\_arr[i] << " ";

    }

    cout << endl;

    return 0;

}

Q3) int main()

{

    int i;

    int arr[5] = {1};  // Only the first element is initialized to 1

    for (i = 0; i < 5; i++)

        printf("%d", arr[i]);

    return 0;

}

// Output: 10000

Q4) #include <iostream>

using namespace std;

int main() {

    int arr[100], n;

    cout << "Enter size of array: ";

    cin >> n;

    cout << "Enter elements: ";

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    // Reverse logic

    for (int i = 0; i < n / 2; i++) {

        int temp = arr[i];

        arr[i] = arr[n - i - 1];

        arr[n - i - 1] = temp;

    }

    cout << "Reversed array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    return 0;

}

Q4) 2.) #include <iostream>

using namespace std;

int main() {

    int A[10][10], B[10][10], C[10][10] = {0};

    int r1, c1, r2, c2;

    cout << "Enter rows and columns of first matrix: ";

    cin >> r1 >> c1;

    cout << "Enter rows and columns of second matrix: ";

    cin >> r2 >> c2;

    if (c1 != r2) {

        cout << "Matrix multiplication not possible!\n";

        return 0;

    }

    cout << "Enter elements of first matrix:\n";

    for (int i = 0; i < r1; i++)

        for (int j = 0; j < c1; j++)

            cin >> A[i][j];

    cout << "Enter elements of second matrix:\n";

    for (int i = 0; i < r2; i++)

        for (int j = 0; j < c2; j++)

            cin >> B[i][j];

    // Matrix multiplication

    for (int i = 0; i < r1; i++) {

        for (int j = 0; j < c2; j++) {

            for (int k = 0; k < c1; k++) {

                C[i][j] += A[i][k] \* B[k][j];

            }

        }

    }

    cout << "Resultant Matrix:\n";

    for (int i = 0; i < r1; i++) {

        for (int j = 0; j < c2; j++)

            cout << C[i][j] << " ";

        cout << endl;

    }

    return 0;

}

Q4) 3.) #include <iostream>

using namespace std;

int main() {

    int A[10][10], T[10][10];

    int rows, cols;

    cout << "Enter rows and columns of matrix: ";

    cin >> rows >> cols;

    cout << "Enter elements:\n";

    for (int i = 0; i < rows; i++)

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

            cin >> A[i][j];

    // Transpose logic

    for (int i = 0; i < rows; i++)

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

            T[j][i] = A[i][j];

    cout << "Transpose of the matrix:\n";

    for (int i = 0; i < cols; i++) {

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

            cout << T[i][j] << " ";

        cout << endl;

    }

    return 0;

}

Q5) #include <iostream>

#include <vector>

using namespace std;

int main() {

    int rows, cols;

    cout << "Enter number of rows and columns: ";

    cin >> rows >> cols;

    vector<vector<int>> matrix(rows, vector<int>(cols));

    cout << "Enter elements of the matrix:\n";

    for (int i = 0; i < rows; ++i) {

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

            cin >> matrix[i][j];

        }

    }

    cout << "\nSum of each row:\n";

    for (int i = 0; i < rows; ++i) {

        int rowSum = 0;

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

            rowSum += matrix[i][j];

        }

        cout << "Row " << i + 1 << " sum: " << rowSum << endl;

    }

    cout << "\nSum of each column:\n";

    for (int j = 0; j < cols; ++j) {

        int colSum = 0;

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

            colSum += matrix[i][j];

        }

        cout << "Column " << j + 1 << " sum: " << colSum << endl;

    }

    return 0;

}