### 1. Program and Algorithm to Insert an Element in an Array

**Algorithm**:

1. Input the array and its size.
2. Input the element to insert and the position.
3. Check if the position is valid.
4. Shift elements to the right to make space for the new element.
5. Insert the new element at the given position.
6. Output the updated array.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[10] = {10, 20, 30, 40, 50};

int n = 5;

int element, position;

cout << "Enter element to insert: ";

cin >> element;

cout << "Enter position to insert (0 to " << n << "): ";

cin >> position;

if (position < 0 || position > n) {

cout << "Invalid position!" << endl;

} else {

for (int i = n; i > position; i--) {

arr[i] = arr[i - 1];

}

arr[position] = element;

n++;

cout << "Array after insertion: ";

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

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

}

cout << endl;

}

return 0;

}

### 2. Program and Algorithm to Delete an Element from an Array

**Algorithm**:

1. Input the array and its size.
2. Input the position of the element to delete.
3. Check if the position is valid.
4. Shift elements to the left to fill the gap.
5. Decrease the array size by 1.
6. Output the updated array.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[10] = {10, 20, 30, 40, 50};

int n = 5;

int position;

cout << "Enter position to delete (0 to " << n - 1 << "): ";

cin >> position;

if (position < 0 || position >= n) {

cout << "Invalid position!" << endl;

} else {

for (int i = position; i < n - 1; i++) {

arr[i] = arr[i + 1];

}

n--;

cout << "Array after deletion: ";

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

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

}

cout << endl;

}

return 0;

}

### 3. Program and Algorithm to Find the Maximum and Minimum Element in an Array

**Algorithm**:

1. Input the array and its size.
2. Initialize variables maxElement and minElement to the first element of the array.
3. Traverse the array to update maxElement and minElement.
4. Output the maximum and minimum element.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[] = {12, 34, 5, 67, 89, 1};

int n = 6;

int maxElement = arr[0], minElement = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] > maxElement) maxElement = arr[i];

if (arr[i] < minElement) minElement = arr[i];

}

cout << "Maximum Element: " << maxElement << endl;

cout << "Minimum Element: " << minElement << endl;

return 0;

}

### 4. Program and Algorithm to Reverse an Array

**Algorithm**:

1. Input the array and its size.
2. Swap the first and last elements, then the second and second-last, and so on.
3. Output the reversed array.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[] = {1, 2, 3, 4, 5};

int n = 5;

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] << " ";

}

cout << endl;

return 0;

}

### 5. Program and Algorithm to Find the Second Largest Element in an Array

**Algorithm**:

1. Input the array and its size.
2. Initialize first and second as the two largest elements.
3. Traverse the array to update first and second.
4. Output the second largest element.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[] = {12, 35, 1, 10, 34, 1};

int n = 6;

int first = arr[0], second = -1;

for (int i = 1; i < n; i++) {

if (arr[i] > first) {

second = first;

first = arr[i];

} else if (arr[i] > second && arr[i] != first) {

second = arr[i];

}

}

if (second == -1) {

cout << "No second largest element!" << endl;

} else {

cout << "Second largest element: " << second << endl;

}

return 0;

}

### 6. Program and Algorithm to Create a 2D Array

**Algorithm**:

1. Input the number of rows and columns.
2. Initialize a 2D array.
3. Input elements for the 2D array.
4. Output the 2D array.

**Program**:

#include <iostream>

using namespace std;

int main() {

int rows, cols;

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

cin >> rows >> cols;

int arr[rows][cols];

cout << "Enter elements: ";

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

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

cin >> arr[i][j];

}

}

cout << "2D Array: " << endl;

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

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

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

}

cout << endl;

}

return 0;

}

### 7. Program to Perform Addition, Multiplication, and Transpose of Two Matrices

**Addition, Multiplication, and Transpose Programs**:

#include <iostream>

using namespace std;

int main() {

int rows = 2, cols = 2;

int matrix1[2][2] = {{1, 2}, {3, 4}};

int matrix2[2][2] = {{5, 6}, {7, 8}};

int result[2][2];

// Addition

cout << "Matrix Addition: " << endl;

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

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

result[i][j] = matrix1[i][j] + matrix2[i][j];

cout << result[i][j] << " ";

}

cout << endl;

}

// Multiplication

cout << "Matrix Multiplication: " << endl;

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

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

result[i][j] = 0;

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

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

cout << result[i][j] << " ";

}

cout << endl;

}

// Transpose

cout << "Matrix Transpose: " << endl;

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

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

cout << matrix1[j][i] << " ";

}

cout << endl;

}

return 0;

}

### 8. Two Sum Problem

**Algorithm**:

1. Input the array and the target sum.
2. Use nested loops to check if any two elements sum up to the target.
3. Output their indices.

**Program**:

#include <iostream>

using namespace std;

int main() {

int nums[] = {2, 7, 11, 15};

int target = 9;

for (int i = 0; i < 4; i++) {

for (int j = i + 1; j < 4; j++) {

if (nums[i] + nums[j] == target) {

cout << "[" << i << ", " << j << "]" << endl;

return 0;

}

}

}

return 0;

}

### 9. Segregating Even and Odd Numbers in an Array

**Algorithm**:

1. Input the array and its size.
2. Use two arrays to store even and odd numbers while maintaining their order.
3. Combine the two arrays and output the result.

**Program**:

#include <iostream>

using namespace std;

int main() {

int arr[] = {12, 34, 45, 9, 8, 90, 3};

int n = 7;

int even[n], odd[n], e = 0, o = 0;

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

if (arr[i] % 2 == 0) {

even[e++] = arr[i];

} else {

odd[o++] = arr[i];

}

}

cout << "Segregated Array: ";

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

cout << even[i] << " ";

}

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

cout << odd[i] << " ";

}

cout << endl;

return 0;

}