

## Array Problem Sheet:

1. Write a program using one dimensional array for the following operations:

- Traversing
- Search
- Sort

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

int main() {
    int arr[100], n, i, j, temp, searchElement, flag = 0;

    cout << "Enter the number of elements: ";
    cin >> n;

    cout << "Enter " << n << " elements: ";
    for (i = 0; i < n; i++) {
        cin >> arr[i];
    }

    // Traversing
    cout << "\nArray elements are: ";
    for (i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    // Searching
    cout << "\n\nEnter element to search: ";
    cin >> searchElement;

    for (i = 0; i < n; i++) {
        if (arr[i] == searchElement) {
            cout << "Element found at position " << i + 1 << endl;
            flag = 1;
            break;
        }
    }
    if (!flag)
        cout << "Element not found!" << endl;

    // Sorting (Bubble Sort)
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
```

```

        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
    }
}
}

// Display sorted array
cout << "\nArray after sorting in ascending order: ";
for (i = 0; i < n; i++) {
    cout << arr[i] << " ";
}
cout << endl;

return 0;
}

```

2. Write a program to display the given two dimensional 3 X3 matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

```

#include <iostream>
using namespace std;

int main() {
    int n, m;
    cout << "Enter the row size: ";
    cin >> n;
    cout << "Enter the column size: ";
    cin >> m;

    int arr1[n][m];
    cout << "Enter the elements of the matrix:\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr1[i][j];
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << arr1[i][j] << " ";
        }
    }
}

```

```

        cout<<"\n";
    }

    return 0;
}

```

3. Perform addition of two matrixes.

**Matrix Addition**

$$\begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 5 \\ 3 & 7 \end{bmatrix} = \begin{bmatrix} 4 & 9 \\ 5 & 8 \end{bmatrix}$$

**Matrix 1**      **Matrix 2**      **Resultant Matrix**

```

#include
<iostream>
using namespace std;

int main() {
    int arr1[2][2], arr2[2][2], add[2][2];
    cout<<"Enter Matrix-1 elements\n";
    for(int i=0;i<2;i++){
        for(int j=0;j<2;j++){
            cin>>arr1[i][j];
        }
    }
    cout<<"Enter Matrix-2 elements\n";
    for(int i=0;i<2;i++){
        for(int j=0;j<2;j++){
            cin>>arr2[i][j];
        }
    }
    cout<<"The Addition matrix is \n";
    for(int i=0;i<2;i++){
        for(int j=0;j<2;j++){
            cout<<arr1[i][j]+arr2[i][j]<< " ";
        }
        cout<<"\n";
    }
    return 0;
}

```

4. Write a program to represent sparse matrix in triplet format.

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

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

int main() {
    int n, m, v_f=0;
    cout << "Enter the row size: ";
    cin >> n;
    cout << "Enter the column size: ";
    cin >> m;

    int arr1[n][m]; // valid after reading n and m

    cout << "Enter the elements of the matrix:\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr1[i][j];
        }
    }

    cout << "\nRows Columns Values\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (arr1[i][j] != 0) {
                v_f+=1;
            }
        }
    }
    cout << n << "    " << m << "    " << v_f << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (arr1[i][j] != 0) {
                cout << i << "    " << j << "    " << arr1[i][j] << endl;
            }
        }
    }

    return 0;
}
```