

1) Develop a Menu driven program to demonstrate the following operations of Arrays

---MENU---

1.CREATE

2.DISPLAY

3.INSERT

4.DELETE

5.LINEAR SEARCH

6.EXIT

```
SOLUTION:- #include <iostream>

using namespace std;

void createArray(int arr[], int &n) {
    cout << "Enter number of elements: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "Enter element " << i + 1 << ": ";
        cin >> arr[i];
    }
}

void displayArray(int arr[], int n) {
    if (n == 0)
        cout << "Array is empty." << endl;
    else {
        cout << "Array elements: ";
        for (int i = 0; i < n; i++)
            cout << arr[i] << " ";
        cout << endl;
    }
}

void insertElement(int arr[], int &n) {
    int pos, val;
    cout << "Enter position (0-based index): ";
    cin >> pos;
    cout << "Enter value to insert: ";
    cin >> val;
```

```

    if (pos < 0 || pos > n) {
        cout << "Invalid position." << endl;
        return;
    }
    for (int i = n; i > pos; i--)
        arr[i] = arr[i - 1];
    arr[pos] = val;
    n++;
    cout << "Element inserted successfully." << endl;
}

void deleteElement(int arr[], int &n) {
    if (n == 0) {
        cout << "Array is empty." << endl;
        return;
    }
    int pos;
    cout << "Enter position (0-based index) to delete: ";
    cin >> pos;
    if (pos < 0 || pos >= n) {
        cout << "Invalid position." << endl;
        return;
    }
    for (int i = pos; i < n - 1; i++)
        arr[i] = arr[i + 1];
    n--;
    cout << "Element deleted successfully." << endl;
}

void linearSearch(int arr[], int n) {
    if (n == 0) {
        cout << "Array is empty." << endl;
        return;
    }
    int key;
    cout << "Enter element to search: ";
    cin >> key;
    for (int i = 0; i < n; i++) {
        if (arr[i] == key) {
            cout << "Element found at position: " << i << endl;
            return;
        }
    }
    cout << "Element not found." << endl;
}

```

```
int main() {
    int arr[100], n = 0, choice;
    while (true) {
        cout << "\n— MENU —" << endl;
        cout << "1. CREATE" << endl;
        cout << "2. DISPLAY" << endl;
        cout << "3. INSERT" << endl;
        cout << "4. DELETE" << endl;
        cout << "5. LINEAR SEARCH" << endl;
        cout << "6. EXIT" << endl;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: createArray(arr, n); break;
            case 2: displayArray(arr, n); break;
            case 3: insertElement(arr, n); break;
            case 4: deleteElement(arr, n); break;
            case 5: linearSearch(arr, n); break;
            case 6: cout << "Exiting program. Goodbye!" << endl; return 0;
            default: cout << "Invalid choice. Try again." << endl;
        }
    }
}
```

```

}
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques1.exe'

ΓÇöΓÇö MENU ΓÇöΓÇö
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: 2
Array is empty.

ΓÇöΓÇö MENU ΓÇöΓÇö
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: 3
Enter position (0-based index): 3
Enter value to insert: 45
Invalid position.

ΓÇöΓÇö MENU ΓÇöΓÇö
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: 4
Array is empty.

```

QUESTION2:- Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

```

CODE:- #include <iostream>

using namespace std;

int removeDuplicates(int arr[], int n) {
    if (n == 0 || n == 1)
        return n;

    int temp[n];
    int j = 0;

```

```

    // Traverse each element
    for (int i = 0; i < n; i++) {
        bool duplicate = false;
        // Check if arr[i] already exists in temp
        for (int k = 0; k < j; k++) {
            if (arr[i] == temp[k]) {
                duplicate = true;
                break;
            }
        }
        if (!duplicate)
            temp[j++] = arr[i];
    }

    // Copy back unique elements
    for (int i = 0; i < j; i++)
        arr[i] = temp[i];

    return j; // Return new size
}

int main() {
    int arr[100], n;

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

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

    n = removeDuplicates(arr, n);

    cout << "Array after removing duplicates: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    cout << endl;
    return 0;
}

```

OUTPUT:-

```
PS C:\Users\Dell\OneDrive\Desktop\ds lab> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques2.exe'
Enter number of elements: 3
Enter array elements: 1 2 3 4 5 6 7 8
Array after removing duplicates: 1 2 3
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output>
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques2.exe'
Enter number of elements: 1 2 2 3 3 4 5 5 5
Enter array elements: Array after removing duplicates: 2
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> |
```

QUESTION3:- 3)Predict the Output of the following program

```
int main()
{
    int i;
    int arr[5] = {1};
    for (i = 0; i < 5; i++)
        printf("%d",arr[i]);
    return 0;
}
```

```
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques3.exe'
10000
OUTPUT:- PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> |
```

QUESTION4-

Implement the logic to

- Reverse the elements of an array

```
SOLUTION:- #include <iostream>
using namespace std;
```

```

void reverseArray(int arr[], int n) {
    int i = 0, j = n - 1, temp;
    while (i < j) {
        temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
    }
}

int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;
    int arr[n];
    cout << "Enter array elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    reverseArray(arr, n);

    cout << "Array after reversing: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;

    return 0;
}

```

OUTPUT:-

```

PS C:\Users\Dell\OneDrive\Desktop\ds lab> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\ques4_a.exe
Enter number of elements: 5
Enter array elements: 1 2 3 4 5
Array after reversing: 5 4 3 2 1
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> 

```

b. Find the matrix multiplication

```

SOLUTION:- #include <iostream>

using namespace std;

int main() {
    int r1, c1, r2, c2;

```

```

int A[10][10], B[10][10], C[10][10];

cout << "Enter rows and columns of Matrix A: ";
cin >> r1 >> c1;
cin.ignore(); // Fix for input skipping

cout << "Enter rows and columns of Matrix B: ";
cin >> r2 >> c2;
cin.ignore(); // Optional but good practice

if (c1 != r2) {
    cout << "Matrix multiplication not possible!" << endl;
    return 0;
}

cout << "Enter elements of Matrix A:\n";
for (int i = 0; i < r1; i++)
    for (int j = 0; j < c1; j++)
        cin >> A[i][j];

cout << "Enter elements of Matrix B:\n";
for (int i = 0; i < r2; i++)
    for (int j = 0; j < c2; j++)
        cin >> B[i][j];

// Matrix multiplication logic
for (int i = 0; i < r1; i++) {
    for (int j = 0; j < c2; j++) {
        C[i][j] = 0;
        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;
}

```



```

PS C:\Users\Dell\OneDrive\Desktop\ds lab> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques4_b.exe'
Enter rows and columns of Matrix A: 1 2
Enter rows and columns of Matrix B: 3 4
Matrix multiplication not possible!
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> 

```

c. Find the Transpose of a Matrix

```

d. #include <iostream>
e. using namespace std;
f.
g. int main() {
h.     int r, c, A[10][10], T[10][10];
i.
j.     cout << "Enter rows and columns of matrix: ";
k.     cin >> r >> c;
l.
m.     cout << "Enter elements of the matrix:" << endl;
n.     for (int i = 0; i < r; ++i)
o.         for (int j = 0; j < c; ++j)
p.             cin >> A[i][j];
q.
r.     // Finding transpose
s.     for (int i = 0; i < r; ++i)
t.         for (int j = 0; j < c; ++j)
u.             T[j][i] = A[i][j];
v.
w.     cout << "Transpose of the matrix:" << endl;
x.     for (int i = 0; i < c; ++i) {
y.         for (int j = 0; j < r; ++j)
z.             cout << T[i][j] << " ";
aa.         cout << endl;
bb.     }
cc.
dd.     return 0;
ee. }
ff.

```

```

PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> cd 'c:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output'
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques4_c.exe'
Enter rows and columns of matrix: 2 2
Enter elements of the matrix:
1 2 3 4
Transpose of the matrix:
1 3
2 4
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output>

```

Question5) Write a program to find sum of every row and every column in a two-dimensional array.

```

SOLUTION:- #include <iostream>

using namespace std;

int main() {
    int r, c;
    cout << "Enter number of rows and columns: ";
    cin >> r >> c;

    int arr[10][10];

    cout << "Enter array elements:" << endl;
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
            cin >> arr[i][j];

    // Sum of each row
    for (int i = 0; i < r; ++i) {
        int row_sum = 0;
        for (int j = 0; j < c; ++j)
            row_sum += arr[i][j];
        cout << "Sum of row " << i + 1 << ": " << row_sum << endl;
    }

    // Sum of each column
    for (int j = 0; j < c; ++j) {
        int col_sum = 0;
        for (int i = 0; i < r; ++i)
            col_sum += arr[i][j];
        cout << "Sum of column " << j + 1 << ": " << col_sum << endl;
    }
}

```

```
    return 0;  
}
```

```
PS C:\Users\Dell\OneDrive\Desktop\ds lab> cd "C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output"  
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> & .\'ques5.exe'  
Enter number of rows and columns: 2    2  
Enter array elements:  
1 2    3 4  
Sum of row 1: 3  
Sum of row 2: 7  
Sum of column 1: 4  
Sum of column 2: 6  
PS C:\Users\Dell\OneDrive\Desktop\ds lab\ass 1\output> 
```