

Ques1.

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

```
int arr[100];
int n;
```

```
void create() {
    cout << "Enter number of elements: ";
    cin >> n;
    if (n > 100) {
        cout << "please try smaller number of elements \n";
        n = 0;
        return;
    }
    cout << "Enter " << n << " elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    cout << "Creted the array \n";
}
```

```
void display() {
    if (n == 0) {
        cout << "Array is empty \n";
        return;
    }
    cout << "Array : ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << "\n";
}
```

```
void insert() {
    if (n == 100) {
        cout << "Array is full. Cannot insert.\n";
        return;
    }
    int pos, val;
    cout << "Enter position of element (1-based index): ";
    cin >> pos;
    if (pos < 1 || pos > n + 1) {
        cout << "wrong position.element cannot be inserted \n";
        return;
    }
    cout << "Enter value: ";
    cin >> val;

    for (int i = n; i >= pos; i--) { // shift elements
        arr[i] = arr[i - 1];
    }
    arr[pos - 1] = val;
```

```

    n++;
}

void deleteEle() {
    if (n == 0) {
        cout << "Array is empty \n";
        return;
    }
    int pos;
    cout << "Enter position to delete (1-based index): ";
    cin >> pos;
    if (pos < 1 || pos > n) {
        cout << "Invalid position.\n";
        return;
    }
    int r = arr[pos - 1];
    for (int i = pos - 1; i < n - 1; i++) { // shift elements left
        arr[i] = arr[i + 1];
    }
    n--;
    cout << "Element " << r << " deleted successfully.\n";
}

void linearSearch() {
    if (n == 0) {
        cout << "Array not found \n";
        return;
    }
    int val;
    cout << "Enter search element ";
    cin >> val;
    for (int i = 0; i < n; i++) {
        if (arr[i] == val) {
            cout << "Element found at position " << i + 1 << "\n";
            return;
        }
    }
    cout << "Element not found.\n";
}

int main() {
    int ch;
    while (true) {
        cout << " MENU \n";
        cout << "1. CREATE\n";
        cout << "2. DISPLAY\n";
        cout << "3. INSERT\n";
        cout << "4. DELETE\n";
        cout << "5. LINEAR SEARCH\n";
        cout << "6. EXIT\n";
        cout << "Enter your choice: ";
        cin >> ch;

        switch (ch) {

```

```

        case 1: create(); break;
        case 2: display(); break;
        case 3: insert(); break;
        case 4: deleteEle(); break;
        case 5: linearSearch(); break;
        case 6: cout << "Exiting program.\n"; return 0;
        default: cout << "Invalid choice \n";
    }
}
}

```

The screenshot shows a C++ IDE with a project named 'dsa_1'. The code editor displays the implementation of a linear search function and a main function with a menu. The output window shows the program's execution, including menu prompts, user choices, and array operations.

```

83 void linearSearch() {
91     for (int i = 0; i < n; i++) {
92         if (arr[i] == val) {
93             cout << "Element found at position " << i + 1 << "\n";
94             return;
95         }
96     }
97     cout << "Element not found.\n";
98 }
99
100 int main() {
101     int ch;
102     while (true) {
103         cout << " MENU \n";
104         cout << "1. CREATE\n";
105         cout << "2. DISPLAY\n";
106         cout << "3. INSERT\n";
107         cout << "4. DELETE\n";

```

Output:

```

6. EXIT
Enter your choice: 1
Enter number of elements: 3
Enter 3 elements: 23 54 67
Creted the array
MENU
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: 2
Array : 23 54 67
MENU
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT

```

Qn2.

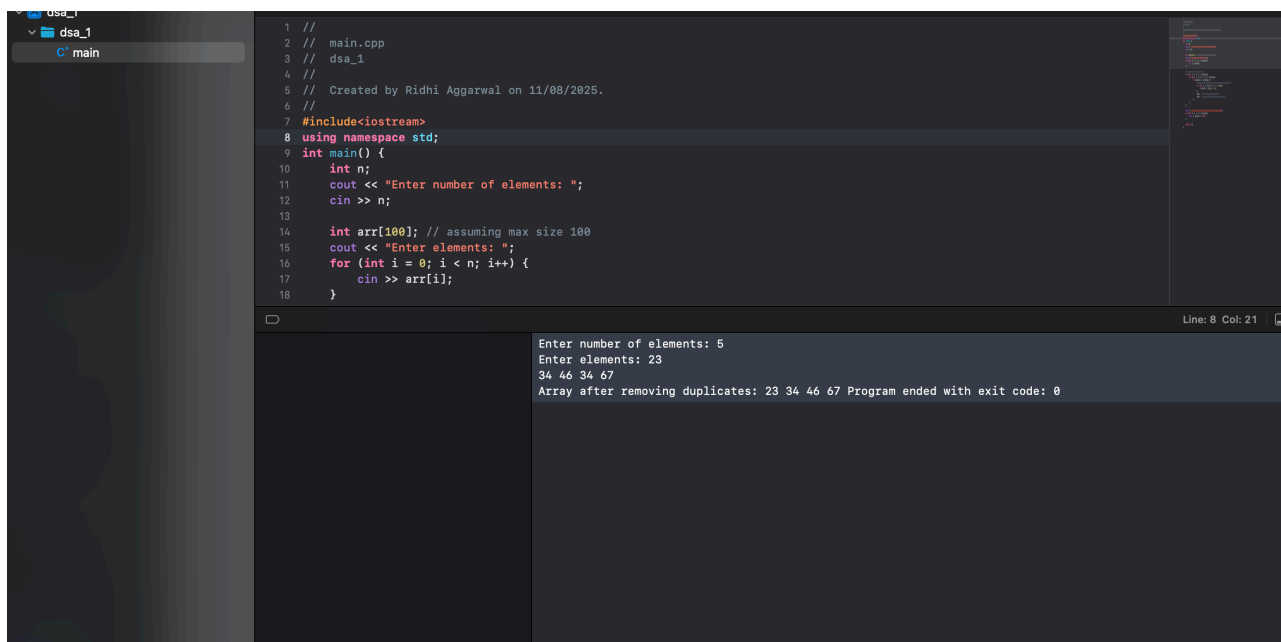
```
int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    int arr[100];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                for (int k = j; k < n - 1; k++) {
                    arr[k] = arr[k + 1];
                }
                n--;
                j--;
            }
        }
    }

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

    return 0;
}
```



The screenshot shows a C++ IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'dsa_1' with a subdirectory 'main'. The code editor contains the same C++ code as shown in the previous block. The terminal output shows the program's execution: it prompts for the number of elements (5), then for the elements (23, 34, 46, 34, 67), and finally displays the array after removing duplicates (23 34 46 67) before ending with exit code 0.

```
1 //
2 // main.cpp
3 // dsa_1
4 //
5 // Created by Ridhi Aggarwal on 11/08/2025.
6 //
7 #include<iostream>
8 using namespace std;
9 int main() {
10     int n;
11     cout << "Enter number of elements: ";
12     cin >> n;
13
14     int arr[100]; // assuming max size 100
15     cout << "Enter elements: ";
16     for (int i = 0; i < n; i++) {
17         cin >> arr[i];
18     }
19
20     for (int i = 0; i < n; i++) {
21         for (int j = i + 1; j < n; j++) {
22             if (arr[i] == arr[j]) {
23                 for (int k = j; k < n - 1; k++) {
24                     arr[k] = arr[k + 1];
25                 }
26                 n--;
27                 j--;
28             }
29         }
30     }
31
32     cout << "Array after removing duplicates: ";
33     for (int i = 0; i < n; i++) {
34         cout << arr[i] << " ";
35     }
36
37     return 0;
38 }
```

Enter number of elements: 5
Enter elements: 23
34 46 34 67
Array after removing duplicates: 23 34 46 67 Program ended with exit code: 0

Ques3.
1000

Ques4.

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

```
void reverseArray(int arr[], int n) {
    cout << "Original Array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
```

```
    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;
}
```

```
void multiplyMatrices() {
    int r1, r2, c1, 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" << endl;
        return;
    }
```

```
    int A[10][10], B[10][10], C[10][10] = {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];
```

```
    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 << " Matrix after multiplication:\n";  
    for (int i = 0; i < r1; i++) {  
        for (int j = 0; j < c2; j++)  
            cout << C[i][j] << " ";  
        cout << endl;  
    }  
}
```

```
void transposeMatrix() {  
    int row, col;  
    cout << "Enter rows and columns of the matrix: ";  
    cin >> row >> col;
```

```
    int mat[10][10], trans[10][10];
```

```
    cout << "Enter matrix elements:\n";  
    for (int i = 0; i < row; i++)  
        for (int j = 0; j < col; j++)  
            cin >> mat[i][j];
```

```
    for (int i = 0; i < row; i++)  
        for (int j = 0; j < col; j++)  
            trans[j][i] = mat[i][j];
```

```
    cout << "Transpose of the Matrix:\n";  
    for (int i = 0; i < col; i++) {  
        for (int j = 0; j < row; j++)  
            cout << trans[i][j] << " ";  
        cout << endl;  
    }  
}
```

```
int main() {  
    int ch;  
    do {  
        cout << " MENU \n";  
        cout << "1. Reverse Array\n";  
        cout << "2. Matrix Multiplication\n";
```

```

cout << "3. Matrix Transpose\n";
cout << "4. Exit\n";
cout << "Enter your choice: ";
cin >> ch;

```

```

switch (ch) {
    case 1: {
        int n;
        cout << "Enter size of array: ";
        cin >> n;
        int arr[100];
        cout << "Enter array elements: ";
        for (int i = 0; i < n; i++) cin >> arr[i];
        reverseArray(arr, n);
        break;
    }
    case 2:
        multiplyMatrices();
        break;
    case 3:
        transposeMatrix();
        break;
    case 4:
        cout << "Exiting program.\n";

```

1
dsa_1
main

```

89  int main() {
112      multiplyMatrices();
113      break;
114      case 3:
115          transposeMatrix();
116          break;
117      case 4:
118          cout << "Exiting program.\n";
119          break;
120      default:
121          cout << "Invalid choice.\n";
122      }
123  } while (ch != 4);
124
125      return 0;
126  }
127

```

MENU

1. Reverse Array
2. Matrix Multiplication
3. Matrix Transpose
4. Exit
Enter your choice: 3
Enter rows and columns of the matrix: 5 3
Enter matrix elements:
12 34 56 78 23 12 23 45 34 23 45 12 16 17 19 90 67 4 83 4 3 2 4 2 34 245 54
Transpose of the Matrix:
12 78 23 23 16
34 23 45 45 17
56 12 34 12 19

MENU

```

                break;
            default:
                cout << "Invalid choice.\n";
            }
        } while (ch != 4);
    return 0;

```

QUES5

```
#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[10][10];
```

```
    // Input matrix elements
```

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

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

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

```
            cin >> arr[i][j];
```

```
        }
```

```
    }
```



```
// Sum of each row

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

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

    int rowSum = 0;

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

        rowSum += arr[i][j];

    }

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

}


// Sum of each column

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

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

    int colSum = 0;

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

        colSum += arr[i][j];

    }

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

}


return 0;
```

dsa_1 › dsa_1 › C:\main › No Selection

```
10 int main() {
32     Sumr += arr[i][j];
33 }
34     cout << "Row " << i + 1 << ": " << Sumr << endl;
35 }
36
37 // Sum of each column
38 cout << "\nSum of each column:\n";
39 for (int j = 0; j < cols; j++) {
40     int colSum = 0;
41     for (int i = 0; i < rows; i++) {
42         colSum += arr[i][j];
43     }
44     cout << "Column " << j + 1 << ": " << colSum << endl;
45 }
46
47     return 0;
48 }
```

filter Results

Enter number of rows: 2
Enter number of columns: 2
Enter elements of the matrix:
23
34 45 15

Sum of each row:
Row 1: 57
Row 2: 60

Sum of each column:
Column 1: 68
Column 2: 49
Program ended with exit code: 0

